TELAC REPORT 91-16A

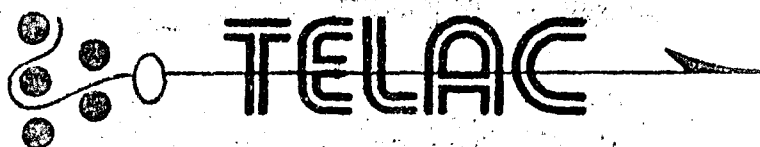# NONLINEAR STALL FLUTTER OF WINGS
WITH BENDING-TORSION COUPLING

AFOSR-TR 92 0783

Final Technical Report for Period: 2 February 1991 - 31 October, 1991

Peter E. Dunn
John Dugundji

DTIC
ELECTE
AUG 17 1992
S   D
B

TELAC

Technology Laboratory for Advanced Composites
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, Massachusetts 02139

DECEMBER, 1991

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
Unclassified	

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	Approved for public release; distribution is unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
TELAC Report 91-16A	

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
Technology Laboratory for Advanced Composites, M.I.T.		AFOSR/NA

6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code)
M.I.T.; Room 33-309 77 Massachuserrs Avenue Cambridge, MA 02139	Bld 410 Bolling AFB, DC 20332-6448

8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR	NA	AFOSR-91-0159

8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS			
Bldg 410 Bolling AFB, DC 20332-6448	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.
	61102F	2302	AS	

11. TITLE (Include Security Classification)
"NONLINEAR STALL FLUTTER OF WINGS WITH BENDING-TORSION COUPLING" (U)

12. PERSONAL AUTHOR(S)
Peter E. Dunn & John Dugundji

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
FINAL TECHNICAL RPT	FROM 2/1/91 TO 10/31/91	31 December 1991	282

16. SUPPLEMENTARY NOTATION
This effort is a continuation of the contract F49620-86-C-0066. The report covers the findings from both of these work units.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Nonlinear flutter, Stall flutter, Composites Aeroelasticity

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The nonlinear, stalled, aeroelastic behavior of rectangular, graphite/epoxy, cantilevered plates with varying amounts of bending-torsion stiffness coupling and with NACA 0012 Styrofoam airfoil shapes is investigated for low Reynolds number flow (<200,000). A general Rayleigh-Ritz formulation is used to calculate point load static deflections, and nonlinear static vibration frequencies and mode shapes for varying tip deflections. Nonlinear lift and moment aerodynamics are used in the context of the Rayleigh-Ritz formulation to calculate static airload deflections. The nonlinear, stalled ONERA model using non-constant coefficients - initially developed by Tran & Petot - is reformulated into a harmonic balance form and compared against a time-marching Runge-Kutta scheme. Low angle-of-attack, linear flutter calculations are done by applying Fourier analysis to extract the harmonic balance method and a Newton-Raphson solver to the resulting nonlinear, Rayleigh-Ritz aeroelastic formulation.

Test wings were constructed and subjected to static, vibration, and wind tunnel tests. Static

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☒ DTIC USERS	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Dr Spencer T. Wu	(202) 767-6962	NA

DD Form 1473, JUN 86   Previous editions are obsolete.

tests indicated good agreement between theory and experiment for bending and torsion stiffnesses. Vibration tests indicated good agreement between theory and experiment for bending and torsion frequencies and mode shapes. 2-dimensional application of the ONERA model indicated good agreement between harmonic balance method and exact Runge-Kutta time integration. Wind tunnel tests showed good agreement between theory and experiment for static deflections, for linear flutter and divergence, and for stalled, nonlinear, bending and torsion flutter limit cycles. The current nonlinear analysis shows a transition from divergence to stalled bending flutter, which linear analyses are unable to predict.

NONLINEAR STALL FLUTTER OF WINGS
WITH BENDING - TORSION COUPLING

Peter E. Dunn
John Dugundji

Technology Laboratory for Advanced Composites
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

December, 1991

FOREWORD

This report describes work done at the Technology Laboratory for Advanced Composites (TELAC) at the Massachusetts Institute of Technology for the Air Force Office of Scientific Research under Grant No. AFOSR-91-0159. Dr. Spencer Wu was the contract monitor.

The work reported herein, was performed during the period, 2 February 1991 through 31 October 1991, and represents a Ph.D. thesis by Peter E. Dunn entitled, "Nonlinear Stall Flutter of Wings with Bending-Torsion Coupling", December 1991, which was completed during this period. This work was a completion of an investigation started earlier under a previous AFOSR contract, No. E49620-86-0066, and was done under the supervision of John Dugundji, the Principal Investigator, and the supporting laboratory staff.

DTIC QUALITY INSPECTED 3

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By.	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

NONLINEAR STALL FLUTTER of
WINGS with BENDING-TORSION COUPLING
by
PETER EARL DUNN

Submitted to the Department of Aeronautics and Astronautics on
December 20, 1991, in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Aeronautics and Astronautics.

ABSTRACT

The nonlinear, stalled, aeroelastic behavior of rectangular, graphite/epoxy, cantilevered plates with varying amounts of bending-torsion stiffness coupling and with NACA 0012 styrofoam airfoil shapes is investigated for low Reynolds number flow (<200,000). A general Rayleigh-Ritz formulation is used to calculate point load static deflections, and nonlinear static vibration frequencies and mode shapes for varying tip deflections. Nonlinear lift and moment aerodynamics are used in the context of the Rayleigh-Ritz formulation to calculate static airload deflections. The nonlinear, stalled ONERA model using non-constant coefficients — initially developed by Tran & Petot — is reformulated into a harmonic balance form and compared against a time-marching Runge-Kutta scheme. Low angle-of-attack, linear flutter calculations are done using the U-g method. Nonlinear flutter calculations are done by applying Fourier analysis to extract the harmonics from the ONERA-calculated, 3-dimensional aerodynamics, then applying a harmonic balance method and a Newton-Raphson solver to the resulting nonlinear, Rayleigh-Ritz aeroelastic formulation.
Test wings were constructed and subjected to static, vibration, and wind tunnel tests. Static tests indicated good agreement between theory and experiment for bending and torsion stiffnesses. Vibrations tests indicated good agreement between theory and experiment for bending and torsion frequencies and mode shapes. 2-dimensional application of the ONERA model indicated good agreement between harmonic balance method and exact Runge-Kutta time integration. Wind tunnel tests showed good agreement between theory and experiment for static deflections, for linear flutter and divergence, and for stalled, nonlinear, bending and torsion flutter limit cycles. The current nonlinear analysis shows a transition from divergence to stalled bending flutter, which linear analyses are unable to predict.

Thesis Supervisor: John Dugundji
Title: Professor of Aeronautics and Astronautics

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a   damping coefficient of ONERA nonlinear aerodynamic differential equation (3-64)

a_0, a_1 parabolic coefficients of damping coefficient of ONERA nonlinear aerodynamic equation

a_{ij} coefficients of polynomial series for i-th region of the aerodynamic force curve

a_{oz} linear slope of the general aerodynamic force coefficient $\equiv dC_{z\ell}/d\alpha$

a_{oL}, a_{oM} linear slopes of lift and moment coefficient curves

A   wing area

A   damping coefficient of ONERA non-linear aerodynamic differential equation (3-64), based only on mean lift deficit Δ_o

[A]   matrix of combined mass and complex aerodynamic components used in linear U-g flutter method

A_{ij} components of combined complex mass-aerodynamic matrix used in linear U-g method

A_{ij} extensional modulus components

AR   aspect ratio $\equiv 2\ell/c$

$[A^{(r,a)}], [A^{(r,e)}]$ aerodynamic matrices resulting from harmonic balance applied to stiffness and damping terms of nonlinear ONERA equation (3-64)

$A_{ij}^{(r,a)}, A_{ij}^{(r,c)}$	components of aerodynamic matrices resulting from harmonic balance applied to ONERA equation (3-64)
b	semi-chord $\equiv c/2$
b_{im}	power series summation in Fourier analysis of the nonlinear aerodynamics, equation (3-84)
B_{ij}	coefficients of i-th torsional mode shape
B_{ij}	extensional-flexural coupling modulus components
c	chord length
$C(k)$	Theodorsen function
C_D, C_L, C_M	drag, lift, and moment coefficients
C_{D_0}	profile drag coefficient
C_{L_α}	slope of linear lift coefficient $\equiv \dfrac{dC_{L\ell}}{d\alpha}$
C_{L2D}	2-dimensional lift coefficient
C_{L3D}	3-dimensional lift coefficient
C_{M2D}	2-dimensional moment coefficient
C_{M3D}	3-dimensional moment coefficient
C_z	general aerodynamic force coefficient
$C_{z\ell}$	static, linear component of general aerodynamic force coefficient
C_{zs}	static general aerodynamic force coefficient
C_{zl}	total linear contribution to the general aerodynamic force coefficient

C_{z2}	nonlinear contribution to the general aerodynamic force coefficient
$C_{z\gamma}$	linear, circulatory contribution to the general aerodynamic force coefficient
C_{zo}, C_{zs}, C_{zc}	mean, sine, and cosine components of the general aerodynamic force coefficient
$C_{z1o}, C_{z1s}, C_{z1c}$	mean, sine, and cosine components of the total linear contribution to the general aerodynamic force coefficient
$C_{z2o}, C_{z2sm}, C_{z2cn}$	mean, sine, and cosine components of the non-linear contribution to the general aerodynamic force coefficient
$C_{z2o}^{(\varphi)}, C_{z2sm}^{(\varphi)}, C_{z2cn}^{(\varphi)}$	mean, sine, and cosine components of the non-linear contribution to the general aerodynamic force coefficient, expressed in the phase shifted domain of $\varphi = k\tau + \xi$
$C_{z\gamma o}, C_{z\gamma s}, C_{z\gamma c}$	mean, sine, and cosine components of the linear, circulatory contribution to the general aerodynamic force coefficient
D_{ij}	flexural modulus components
e	phase lag of ONERA nonlinear aerodynamic differential equation (3-64)
e_0, e_1	parabolic coefficients of phase lag of ONERA nonlinear aerodynamic equation
E	phase lag of ONERA nonlinear aerodynamic differential equation (3-64), based only on mean lift deficit Δ_o
E_L	longitudinal modulus

E_T	transverse modulus
EA	extensional stiffness
EAe_A^*	extensional/bending coupling stiffness
$EB_2^{**}, EB_3^{**}, EB_4^{**}$	nonlinear stiffness terms, from Section 3.1.2
EC_1	torsional warping stiffness
EC_1^{**}	torsion/fore-&-aft coupling stiffness
EI_η	out-of-plane bending stiffness
EI_ζ	fore-&-aft bending stiffness
EI_{16}	out-of-plane/torsion coupling stiffness
f	residual vector in Newton-Raphson scheme
f, f_j	cosh and sinh parameters of torsional mode shapes
$F(k)$	real part of Theodorsen function
g	structural damping in linear U-g flutter method
g, g_j	cosine and sine parameters of torsional mode shapes
G_{LT}	shear modulus
GJ_x	torsional stiffness
$G(k)$	imaginary part of Theodorsen function
h	1/4-chord tip deflection
$\bar{h}$	non-dimensional 1/4-chord deflection
$\bar{h}_0, \bar{h}_s, \bar{h}_c$	mean, sine, and cosine components of the non-dimensional 1/4-chord deflection

H_{mij}	torsion/fore-&-aft nonlinear coupling stiffness term due to m-th bending mode shape
H.H.T	abbreviation for "Higher Harmonic Terms"
H.O.T.	abbreviation for "Higher Order Terms"
i_x	mass moment of inertia
I_1, I_2	modulus invariants
I_{ij}	mode shape integrals corrected for spanwise effects, used in linear U-g flutter method
I_{im}	sinusoidal integrals in Fourier analysis of the nonlinear aerodynamics, equation (3-85)
$[I']$	diagonal matrix of squares of harmonic frequencies in harmonic balance method
J_i	maximum power of the polynomial series for the i-th region of the aerodynamic force curve
k	reduced frequency $\equiv \omega b/U$
k_A	radius of gyration of cross section
k_{vz}	second apparent mass coefficient
k_{vL}, k_{vM}	lift and moment second apparent mass coefficients
[K]	stiffness matrix
K_{ij}	components of the stiffness matrix
K_{ij}^{fp}, K_{ij}^{sty}, K_{ij}^{geo}	flat plate, styrofoam, and geometric contributions to the components of the stiffness matrix
ℓ	semi-span, laminate length

L_1, L_2, L_3, L_4	components of complex lift used in linear U-g flutter method
L_s, L_c	intermediate variables of the harmonic balance approximation applied to the ONERA linear differential equation
L_v, L_w	generalized forces in y and z directions
m	mass per unit area
$[M]$	mass matrix
M_1, M_2, M_3, M_4	components of complex moment used in linear U-g flutter method
M_{ij}	components of the mass matrix
$M_{ij}^{fp}, M_{ij}^{sty}, M_{ij}^{geo}$	flat plate, styrofoam, and geometric contributions to the components of the mass matrix
M_η	bending moment resultant about η axis
M_ζ	bending moment resultant about ζ axis
M_θ	generalized applied moment about x axis
n	number of mode shapes
n_b, n_t, n_c, n_f	number of bending, torsion, chordwise bending, and fore-&-aft mode shapes
P_x	warp term in strain energy equation
q_i	i-th modal amplitude
q_{io}, q_{is}, q_{ic}	mean, sine, and cosine components of the i-th modal amplitude
$\{q_o\}, \{q_s\}, \{q_c\}$	vectors of the means, sines, and cosines of the modal amplitudes

$\tilde{q}$	frequency domain modal amplitude, used for free vibration analysis and linear flutter analysis
Q_i	i-th modal force
Q_{io}, Q_{is}, Q_{ic}	mean, sine, and cosine components of the i-th modal force
$\{Q_o\}, \{Q_s\}, \{Q_c\}$	vectors of the means, sines, and cosines of the modal forces
Q_{ij}	unidirectional modulus components
$Q_{ij}^{(\theta)}$	rotated modulus components
r	stiffness term of ONERA nonlinear aerodynamic differential equation (3-64)
r_0, r_1	parabolic coefficients of stiffness term of ONERA nonlinear aerodynamic equation
R	stiffness term of ONERA nonlinear aerodynamic differential equation (3-64), based only on mean lift deficit Δ_o
R_{mnij}	torsion/torsion nonlinear coupling stiffness term due to m-th and n-th bending mode shapes
R_1, R_2	modulus invariants
s_z	first apparent mass coefficient
s_L, s_M	lift and moment first apparent mass coefficients
S_x	twisting moment in strain energy equation arising from shear stress
t	real time
t_{fp}	thickness of graphite/epoxy flat plate

t_{max}	maximum thickness of styrofoam NACA airfoil shape
T	kinetic energy
T_x	twisting moment in strain energy equation arising from logitudinal stress
U	strain energy
U	free stream velocity
v	fore-&-aft deflection
V_x	axial force (tension) in the spanwise x-direction
w	out-of-plane deflection
x	spanwise cartesian coordinate
x	state vector in Newton-Raphson scheme
x_{sty}	spanwise location at which styrofoam fairing starts
y	chordwise cartesian coordinate
z	out-of-plane cartesian coordinate
z_k	distance from mid-plane to upper surface of k-th ply
Z	complex eigenvalue of linear U-g flutter equation
α	effective angle-of-attack
$\alpha_o, \alpha_s, \alpha_c$	mean, sine, and cosine components of the effective angle-of-attack
α_c	angle-of-attack corrected for 3-dimensional finite span effects

α_i	angle delineating the beginning of the i-th polynomial region of the approximation to the aerodynamic force curve
α_j	sinh and sine coefficients of the j-th beam bending mode shape
α_L, α_M	lift and moment phase lags of ONERA linear aerodynamic differential equation
α_R	root angle-of-attack, same as θ_R
α_v	oscillatory amplitude of the effective angle-of-attack $\equiv \sqrt{\alpha_s^2 + \alpha_c^2}$
α_z	phase lag of the ONERA linear aerodynamic differential equation (3-63)
α_Δ	stall angle for single break point model
β	non-dimensional bending-torsion stiffness ratio
β_1, β_2, β_3	coefficients of nonlinearity of harmonically balanced, nonlinear ONERA equation
γ_i^v, γ_i^w	fore-&-aft and out-of-plane components of the i-th mode shape
ΔC_z	deviation of the non-linear aerodynamic force curve from the linear approximation
ΔC_{zv1}, ΔC_{zv2}	first & second harmonic oscillatory amplitudes of the deviation of the non-linear aerodynamic force curve from the linear approximation
ε_j	parameter of j-th beam bending mode shapes
ε_{xx}, $\varepsilon_{\eta\eta}$, $\varepsilon_{x\eta}$, $\varepsilon_{x\zeta}$, $\varepsilon_{\eta\zeta}$	engineering strain components

ζ	through-the-thickness or out-of-plane direction in local coordinates
η	chordwise direction in local coordinates
θ	ply angle referenced to free stream
θ	instantaneous angle-of-attack
$\theta_o, \theta_s, \theta_c$	mean, sine, and cosine components of the instantaneous angle-of-attack
θ_R	root angle-of-attack, same as α_R
$\lambda(\eta,\zeta)$	warp function
λ_z	lag of ONERA linear aerodynamic differential equation (3-63)
λ_L, λ_M	lift and moment lags of ONERA linear aerodynamic differential equation
ν_{LT}	Poisson's ratio
ξ	phase of effective angle-of-attack $\approx \tan^{-1}\dfrac{\alpha_c}{\alpha_s}$
ρ	free stream density
ρ_j	parameter of j-th beam bending mode shape
ρ_s	styrofoam density
$\sigma_{xx}, \sigma_{\eta\eta}, \sigma_{x\eta},$ $\sigma_{x\zeta}, \sigma_{\eta\zeta}$	engineering stress
σ_z	linear phase lag used in equation (3-63)
σ_L, σ_M	lift and moment linear phase lags
τ	non-dimensional time $\approx \dfrac{Ut}{b}$

ϕ_i^v, ϕ_i^w spanwise variations of fore-&-aft and out-of-plane components of i-th mode shape

$\psi_i^v(z)$ through-the-thickness variation the fore-&-aft component of the i-th mode shape

$\psi_i^w(y)$ chordwise variation the out-of-plane component of the i-th mode shape

ω real frequency

φ_i non-dimensional phase delineating the beginning of the i-th polynomial region of the approximation to the aerodynamic force curve

Chapter I
Introduction

The analysis of aircraft flutter behavior is traditionally based on small amplitude, linear theory, in regards to both structural and aerodynamic modeling. However, if the wing is near the stall region, a nonlinear stall flutter limit cycle may occur at a lower velocity than linear theory would suggest. Moreover, near the divergence velocity, large deflections producing angles of attack near the stall angle may also trigger a flutter response. Since some current aircraft are achieving high angle of attack for maneuvering, and since rotorcraft may use long, highly-flexible blades for their rotors, it is of interest to investigate both this nonlinear stall flutter behavior and this large amplitude deflection behavior, and their transitions from linear behavior. The development of advanced composite materials allows the aircraft designer another parameter by which he might control these new behaviors — his ability to control the anisotropy of advanced composite materials through selective lamination makes these materials attractive for aeroelastic tailoring.

The present research is part of a continuing investigation at the Technology Laboratory for Advanced Composites at M.I.T. into the aeroelastic flutter and divergence behavior of forward-swept, graphite/epoxy composite wing aircraft. The specific objectives of the current investigation are to explore experimentally and analytically the roles of nonlinear structures and nonlinear aerodynamics in high angle-of-attack stall flutter of aeroelastically tailored wings, while attempting to develop a nonlinear method of analysis that is

not overly computationally intensive, i.e. that is suitable for routine aeroelastic analysis.

Chapter 2 describes some of the previous work and analytic approaches used to grapple with the problem of stall flutter of composite wings. This chapter includes a description of some of the previous work at TELAC that has concentrated on the beneficial effects of the bending-torsion coupling of composite wings, but that has mostly been relegated to linear analysis. It also describes preliminary work in the current investigation that sets up some of the analytic models that have been chosen to approach the stall flutter problem.

Chapter 3 describes the theory involved in the current work that seeks to expand on and improve the efforts of the previous investigations, described in Chapter 2. Analytically, it was endeavored to more accurately model the nonlinearities over the preliminary investigation: aerodynamic nonlinearities were incorporated in both the forcing terms and in the equations of motion; structural nonlinearities were developed analytically from geometric considerations.

Chapter 4 describes the experiments performed so as to corroborate the theoretical analysis. As with the previous work, static tests and vibration tests were employed to verify mass and stiffness properties. Experimentally, the wings were designed so as to better allow an investigation of the linear-to-nonlinear transition, while also improving the Reynolds number range. The experimental procedure was also modified so as to acquire more data on larger amplitude flutter oscillation.

Chapters 5 and 6 detail the products of the theoretical and experimental investigations, comparing the results of the two, with concluding remarks on the significant contributions of the current investigation and recommendations for further work.

4

Chapter II
Summary of Previous Work

2.1 Dynamic Stall Models

Much work has been done in creating a large base of dynamic stall experimental data for airfoils in sinusoidal pitch motion, from which might be developed models to analytically reproduce their behavior. The intent of this experimental work was to observe the 2-dimensional dynamic stalling behavior of various airfoils while varying a large number of parameters — such as airfoil shape, mean angle of attack, amplitude of oscillation of angle of attack, reduced frequency, Mach number, Reynolds number, leading edge geometry, et cetera. Initial work was done Liiva & Davenport [Ref. 1], with discussion of the effects of Mach number. Extensive work was done by McAlister, Carr, & McCroskey [Ref. 2] in producing a data base for the NACA 0012 airfoil, and extended by McCroskey, McAlister, Carr, Pucci, Lambert, & Indergrand [Ref. 3] and by McAlister, Pucci, McCroskey, & Carr [Refs. 4 and 5] to include other airfoil shapes and a wider range in the variable parameters. The general conclusion of these experiments was that the parameters of the unsteady motion itself appear to be more important than airfoil geometry — however, most of these experiments were conducted for deep dynamic stall, i.e. vortex-dominated cases. Light dynamic stall cases, which are less severe and more common for practical applications, appear to depend on all the parameters of the unsteady motion.

Coincidentally with these experiments, attempts were made to identify the processes that make up the dynamic stall event. With the aid of chordwise propagation of pressure waves [Ref. 6], flow visualization [Ref. 7 and 8], and data from hot-wire probes and surface pressure transducers [Ref. 2], Carr, McAlister, & McCroskey [Refs. 9 and 10] identified the characteristic processes illustrated in Fig. 1 [Fig. 27 from Ref. 9]. However, it should be noted that the NACA 0012 airfoil exhibits trailing-edge stall — i.e. the dynamic stall phenomenon originates from an initial boundary layer separation at the trailing edge — while other airfoil shapes might exhibit leading-edge or mixed stall behavior.

Because of the prevalence of dynamic stall in rotorcraft, where the drop in dynamic pressure for a retreating blade might necessitate angles of attack beyond the stall angle so as to maintain lift, appropriate modeling of the dynamic stall phenomenon has been a primary concern in helicopter design for over two decades. Research in this area has followed two approaches, one theoretical [Refs. 11 to 30], and the other based on experimental data, also called semi-empirical [Refs. 31 to 59]. These research efforts are well summarized, and their advantages and disadvantages compared, in Refs. 10 and 60-65.

The theoretical approaches are the discrete potential vortex approach, zonal methods, and Navier-Stokes calculations. The discrete potential vortex approach [Refs. 11 to 17] ignores the viscous terms in the fundamental equations and assumes potential flow without the boundary layer. This type of model takes its cue from
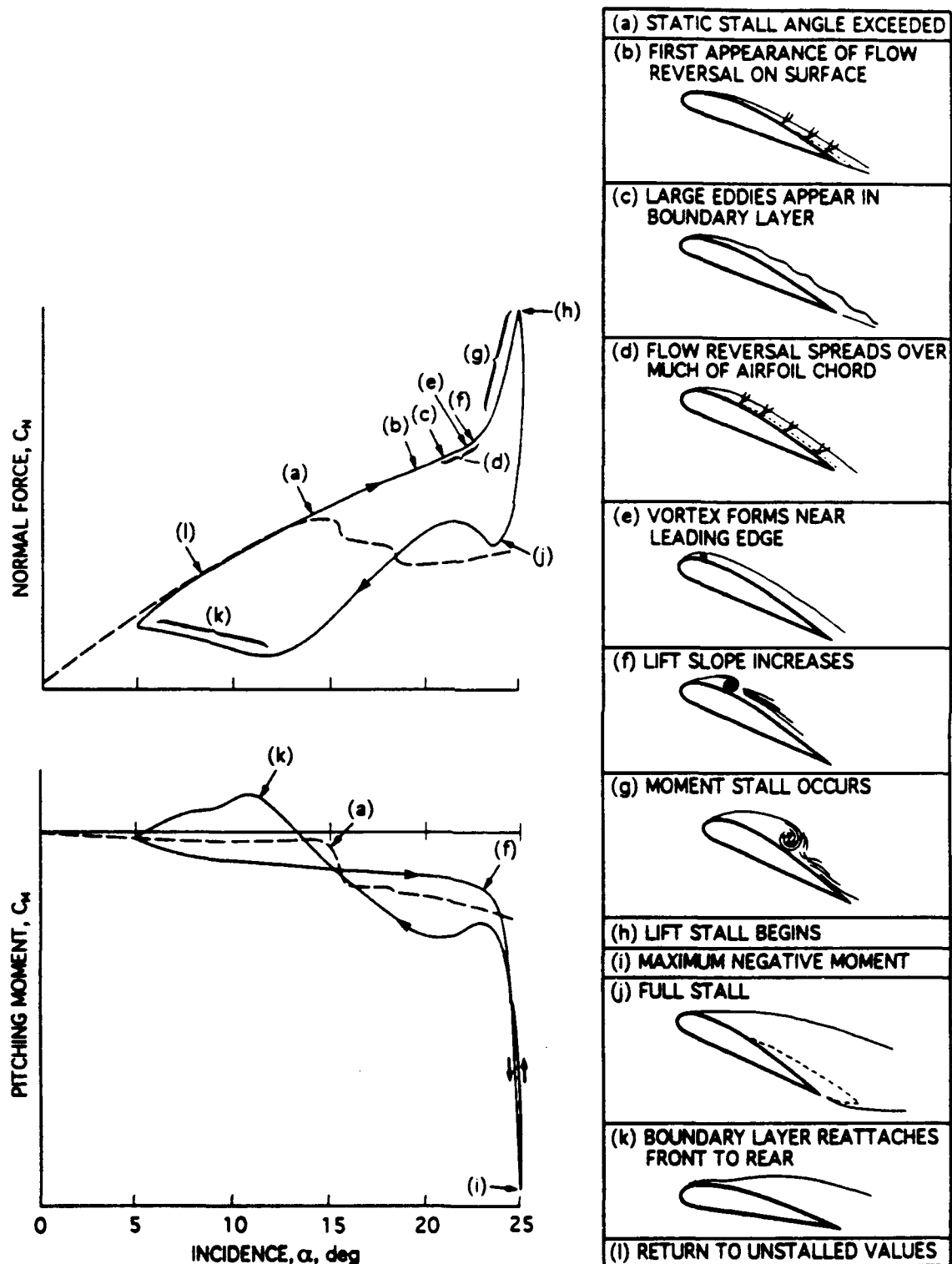
Fig. 1 Dynamic Stall Events on NACA 0012 Airfoil

7

the discrete vortex model that has been applied to bluff body separation — the viscous part of the flow is taken into account by the generation and transport off the leading and trailing edges of discrete combined vortices, governed by semi-empirical or boundary layer considerations. Zonal methods [Refs. 18 to 23] model separately the viscous, nonviscous, and transition regions of the flow, under the assumption that the viscous region usually remains relatively thin. The limitations and approximations of the discrete vortex and zonal methods can, in principle, be avoided by solving the full Navier-Stokes equations [Refs. 24 to 30]. However, turbulence must be modeled — many solutions incorporate the so-called Reynolds-averaged, Navier-Stokes equations so that the Reynolds stress, which vanishes in laminar flow, can be modeled in the turbulent case.

These three theoretical models are computationally intensive and are limited by the approximations and restrictions of their formulations, thus usually making them unsuitable for routine aeroelastic analysis.

The semi-empirical methods attempt to use static data with corrections for the dynamic nature of the dynamic stall event, choosing to model only the gross characteristics of the phenomenon while ignoring the fine details of the fluid flow. This is advantageous because the static data already takes into account the effects of Reynolds number, Mach number, and airfoil shapes, and because the methods are therefore not as computationally intensive as the theoretical methods, thus making them more suitable for routine aeroelastic analysis. These semi-empirical methods are the Boeing-Vertol

gamma function method, the UTRC or UARL method, the MIT Method, the Lockheed method, time-delay methods, and the ONERA method.

The Boeing-Vertol gamma function method [Refs. 31 to 33] uses a corrected angle of attack — calculated as a function of the rate of change of the angle of attack — when the angle exceeds the static stall angle, based on γ, the essential empirical function of airfoil geometry and Mach number. The UTRC or UARL α, A, B method [Refs. 34 to 36], developed at United Technologies Research Center, is a table-lookup correlation method based on a 3-dimensional array of measured data (angle of attack, reduced pitch rate, reduced pitch acceleration), and therefore requires a large amount of data storage for each airfoil, frequency of oscillation, and the associated interpolation. Recent advances have been made on the UTRC method [Refs. 37 to 39] to reduce these large volumes of data into compact expressions (synthesization). The MIT method [Refs. 40 and 41], like the Boeing-Vertol method, corrects the angle of attack as a function of its rate, but empirically represents the forces due to the vortex shedding phenomenon for ramp changes in angle of attack, such that they increase linearly to the peak C_L and C_M values observed from ramp experiments. The Lockheed method [Refs. 42 to 45] is a combined analytical and empirical modeling that incorporates phase-lag time constants and pitch-rate-dependent, stall-angle delay increments into an effective angle of attack, together with a number of separate dynamic stall elements — based on analogy to other dynamic and/or turbulent flow phenomenon — to construct the aerodynamic forces. Time delay methods [Refs. 46 to 48] assume that

each dynamic stall event is governed by a universal, dimensionless time constant, regardless of the time history of the motion.

Finally, the ONERA method [Refs. 49 to 55], developed by Tran, Petot, & Dat of *Office National d'Etudes et de Recherches Aerospatiales*, uses a second-order differential equation with non-constant coefficients to model the deviation of the dynamic stall behavior from that of the theoretical linear behavior, with a fixed-time stall delay, $\Delta \tau$ (usually 5 or 10). The parameters/coefficients of the differential equations are derived empirically, usually from small amplitude-of-oscillation experiments, and are meant to reflect the frequency and damping of the dynamic stall processes. Petot & Loiseau [Ref. 51] indicate how the ONERA method might be adapted for Reynolds numbers below the critical Re value. Petot [Ref. 52] demonstrates how the coefficients of the differential equations might be derived from a few large-amplitude-of-oscillation cases, instead of a large number of small-amplitude-of-oscillation cases, thus taking advantage of a smaller data base of such types of experiments. McAlister, Lambert, & Petot [Ref. 53] demonstrate a systematic procedure for determining the empiric parameters, approaching the problem from an engineering point of view. Petot & Dat [Ref. 54] reformulate the differential equations so that they reduce to the Theodorsen and Kussner functions in the case of a flat plate in the linear domain.

Some work has been done to extend these empirical methods from purely sinusoidal pitching motion to pitch & plunge motion [Refs. 55 to 59]. In particular, Peters [Ref. 55] and Rogers [Ref. 56] present physical arguments for the manner in which the pitching and

10

plunging motions should be separated in the ONERA differential equations. In general, these empirical methods are employed is some type of stripwise theoretical fashion, since little experimental or analytical work has been done on the 3-dimensional effects of dynamic stall.

2.2 Structural Models

For a flutter analysis, it is first necessary to correctly describe the linear structural equations of motion of the wing. In general, this entails accurately modeling the linear frequencies and mode shapes, since linear flutter usually involves the coalescence of modes, while nonlinear stall flutter usually involves single degree-of-freedom behavior.

The modeling of bending and torsion modes for uniform beams and plates is already well established [Refs. 66 and 67]. However, for plates which are uniform along the span but anisotropic in nature, the analytic tools have only recently been developed because of the relative newness of composite materials. Crawley, Dugundji, & Jensen [Refs. 68 to 71] have set up the appropriate equations of motion and have determined the types and number of modes to accurately evaluate the natural frequencies and mode shapes of composite plates.

Several approaches have been taken to account for the geometric, structural nonlinearities that can become important for aeroelastic analysis with large deflections. Some work has been done using the Finite Element method with application to rotor blades [Refs. 72 and 73]. Dugundji & Minguet [Ref. 74] have developed a

model based on Euler angles which can account for arbitrarily large deflections, and in which the equations of motion are solved by a Finite Difference scheme.

However, many of the approaches for long, flexible blades involve ordering schemes, which rely on being able to identify non-linear terms of various orders, and truncating the equations of motion accordingly [Refs. 75 to 77]. Such a formulation, by Hodges & Dowell [Ref. 75], can be implemented for an analysis where it is assumed that out-of-plane bending is moderate in amplitude, while torsion and fore-&-aft bending are relatively small. This model derives the equations of motion by Hamilton's principle for long, straight, slender, homogeneous, isotropic beams, and is valid to second order. Its ordering scheme assumes that the squares of the bending slopes, the torsion deformation, and the chord/radius and thickness/radius ratios are negligible with respect to unity. The equations can also be converted into a modal formulation, as has been done by Boyd [Ref. 78]. However, little work has been done to modify these nonlinear structural equations to account for the anisotropy of composite materials.

Other nonlinear work, such as that by Tseng & Dugundji [Refs. 79 and 80], has noted the often encountered, cubic stiffening phenomenon of many nonlinear, structural vibration problems.

2.3 Stall Flutter Analysis

The characteristics of — and factors affecting — stall flutter have been identified in early work by Halfman, Johnson, & Haley [Ref. 81] and by Rainey [Refs. 82 and 83]: (i) there is a sharp drop

in the critical flutter speed, (ii) the flutter frequency rises toward the torsional frequency, and (iii) the motion is predominantly torsional, i.e. single degree of freedom flutter. These characteristics are quite distinct from those of classical linear flutter where the unsteady instability is generated from the coalescence of bending and torsion modes and frequencies. Additional experimental work has been done by Dugundji, et al. [Refs. 84 and 85] to investigate the 2-dimensional, large-amplitude, stall flutter behavior of a flat plate with a linear torsional spring.

Weisshaar, et al. [Refs. 86 to 89] have concentrated on the aeroelastic advantages of using composite materials. This work has investigated the parameters of layup, sweep, taper, aspect ratio, etc. for such applications as flutter & divergence suppression, lift effectiveness, control effectiveness, and mode shape & frequency tailoring. In general, the models used were 2D strip theory for aerodynamics, and a comparison of high aspect ratio plate, chordwise rigid, and laminated tube models for structures.

Recent work at M.I.T. by Dugundji, et al. [Refs. 90 to 92] has concentrated on taking advantage of bending-torsion coupling for flutter modeling. These investigations at the Technology Laboratory for Advanced Composites (TELAC) looked at the aeroelastic flutter and divergence behavior of cantilevered, unswept and swept, graphite/epoxy wings in a small, low-speed wind tunnel. The wings were six-ply, graphite/epoxy plates with strong bending-torsion coupling. Experiments were conducted to determine the flutter boundaries of these wings both at low and high angles of attack, stall flutter often being observed in the latter. Hollowell & Dugundji

13

[Ref. 90] presented the first of these aeroelastic investigations, with linear structures and V-g linear flutter analysis applied as strip theory. Selby [Ref. 91] extended this same aeroelastic analysis by applying a doublet lattice aerodynamic model. Landsberger & Dugundji [Ref. 92] further extended this analysis to include wing sweep, with the 3D Weissinger L-method for steady aerodynamics. The divergence and flutter results at low angles of attack correlated well with linear, unsteady theory, indicating some beneficial effects of ply orientation in aeroelastic behavior. Steady, nonlinear aerodynamics correlated reasonably before the onset of flutter, but none of these previous analyses attempted to tackle the nonlinearities that occurred due to dynamic stalling and large amplitude deflections.

Harmonic balance methods have been used as a means to approach such nonlinear problems [Ref. 93]. While these methods do not model the fine details of the nonlinear motion, as would time marching schemes, they are suitable for describing the gross aspects of the solution if the nonlinearity is sufficiently moderate. Therefore, they seem particularly suited to stall flutter analyses, since most of the semi-empirical aerodynamic models likewise choose to ignore the fine details of the fluid flow.

Most recently the work at M.I.T. has been extended by Dunn & Dugundji [Refs. 94 and 95] to investigate the nonlinearities in the flutter behavior of composite wings — this constituted a preliminary effort toward the current investigation. The ONERA, semi-empirical, aerodynamic model was applied in a 2D stripwise fashion, with empirical corrections for Reynolds number and 3D effects. However, the aerodynamic nonlinearities were modeled in the aerodynamic

forcing terms only, i.e. the nonlinear effects on the natural frequency and damping of the stalled behavior were not modeled in a time varying fashion and would therefore break down for large amplitudes of oscillation. The structural model was linear, with empirical corrections for cubic stiffening, and the combined equations of motion were reduced algebraically by a harmonic balance method.

Corresponding to this analytic work, experimental work was conducted to verify these analytic models. Experimental static tests and vibration tests were conducted to verify the mass and stiffness properties of the wings. Small-amplitude flutter experiments were conducted to corroborate the analytic flutter model. However, the linear flutter velocity of the wings was above the wind tunnel velocity, precluding experimental investigation of transition from pure linear to stalled, nonlinear behavior. Also, while the analytic model existed to investigate larger amplitude flutter oscillation, little of such data was taken experimentally.

The principal contributions of this preceding work were the reduction by harmonic balance and Fourier analysis of some of the parameters of the nonlinear ONERA equations; the analytic investigation of some of the single degree of freedom, stall flutter phenomena; and the preliminary development of an experimental base of data for stall flutter of composite wings. However, this initial work fails to incorporate any analytic, structural nonlinearities; ignores some of the salient features of the ONERA equations in its application of the harmonic balance method; and requires further accumulation of large amplitude of oscillation data.

16

Chapter III
Theory

3.1 Structural Model

3.1.1 Anisotropic Plate Modulus Components

The flexural modulus components of a laminated, graphite/epoxy plate depends on both the fiber orientations and stacking sequence of the individual plies. Only laminated plates with mid-plane symmetric stacking sequences were constructed in this study. The ply angles (θ) follow the sign convention in Fig. 2.

The in-plane, unidirectional modulus components Q_{ij} were obtained from the orthotropic engineering constants for Hercules AS4/3501-6 graphite/epoxy, from which the test specimens were fabricated. These engineering constants take on different values depending on whether they are obtained from out-of-plane bending or in-plane stretching tests. Engineering constants obtained from each type of test appear in Appendix A, and the out-of-plane values were used in the current analysis because in-plane stretching was assumed to be negligible. The Q_{ij} terms are defined in terms of the engineering constants as,

$$(3-1) \qquad Q_{11} = \frac{E_L}{1 - \nu_{LT}\nu_{TL}}$$

$$(3-2) \qquad Q_{22} = \frac{E_T}{1 - \nu_{LT}\nu_{TL}}$$
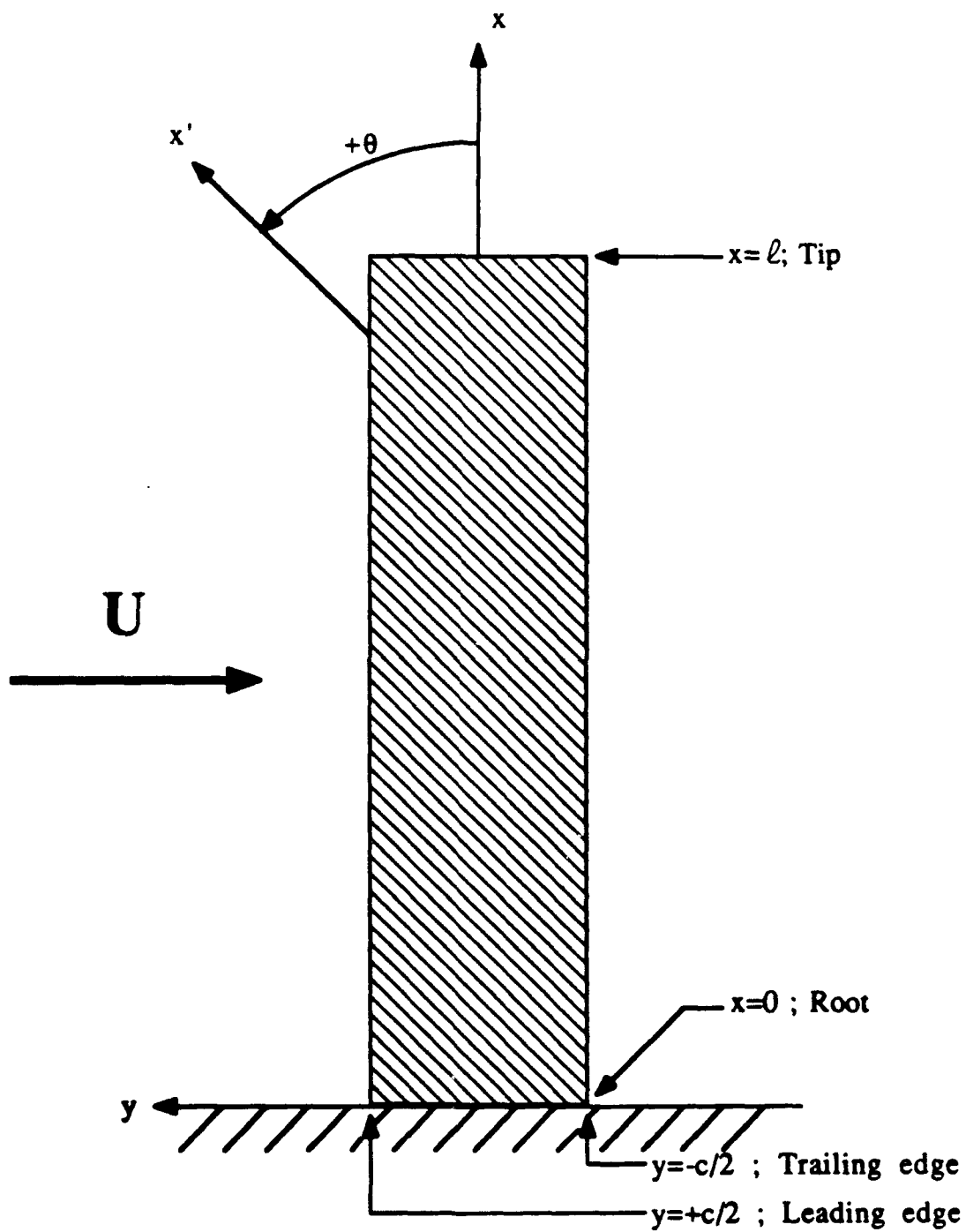
Fig. 2 Sign convention for ply angles and axes

$$(3\text{-}3) \qquad Q_{12} = Q_{21} = \frac{\nu_{LT}E_L}{1 - \nu_{LT}\nu_{TL}}$$

$$(3\text{-}4) \qquad Q_{66} = G_{LT}$$

where,

$$(3\text{-}5) \qquad \nu_{TL} = \frac{E_T}{E_L}\nu_{LT}$$

The in-plane, rotated modulus components were obtained by first defining a set of invariants,

$$(3\text{-}6) \qquad I_1 = \frac{1}{4}[Q_{11} + Q_{22} + 2Q_{12}]$$

$$(3\text{-}7) \qquad I_2 = \frac{1}{8}[Q_{11} + Q_{22} - 2Q_{12} + 4Q_{66}]$$

$$(3\text{-}8) \qquad R_1 = \frac{1}{2}[Q_{11} - Q_{22}]$$

$$(3\text{-}9) \qquad R_2 = \frac{1}{8}[Q_{11} + Q_{22} - 2Q_{12} - 4Q_{66}]$$

The invariants are transformed to the rotated modulus components using the relations,

$$(3\text{-}10) \qquad Q_{11}^{(\theta)} = I_1 + I_2 + R_1\cos2\theta + R_2\cos4\theta$$

$$(3\text{-}11) \qquad Q_{22}^{(\theta)} = I_1 + I_2 - R_1\cos2\theta + R_2\cos4\theta$$

$$(3\text{-}12) \qquad Q_{12}^{(\theta)} = I_1 - I_2 - R_2\cos4\theta$$

19

$$(3-13) \qquad Q_{66}^{(\theta)} = I_2 - R_2 \cos 4\theta$$

$$(3-14) \qquad Q_{16}^{(\theta)} = \frac{1}{2} R_1 \sin 2\theta + R_2 \sin 4\theta$$

$$(3-15) \qquad Q_{26}^{(\theta)} = \frac{1}{2} R_1 \sin 2\theta - R_2 \sin 4\theta$$

where θ is the ply angle.

The flexural modulus components, D_{ij}, for an n-ply laminate with arbitrary ply angle orientation are obtained from,

$$(3-16) \qquad A_{ij} = \sum_{k=1}^{n} Q_{ij}^{(\theta)}(z_k - z_{k-1}) \quad i,j = 1,2,6$$

$$(3-17) \qquad B_{ij} = \sum_{k=1}^{n} Q_{ij}^{(\theta)} \frac{z_k^2 - z_{k-1}^2}{2} \quad i,j = 1,2,6$$

$$(3-18) \qquad D_{ij} = \sum_{k=1}^{n} Q_{ij}^{(\theta)} \frac{z_k^3 - z_{k-1}^3}{3} \quad i,j = 1,2,6$$

where,

θ_k = ply angle of the k-th ply

z_k = distance from the mid-plane to the upper surface of the k-th ply (positive above mid-plane, negative below mid-plane)

z_{k-1} = distance from the mid-plane to the lower surface of the k-th ply

20

3.1.2 Equations of Motion with Geometric Nonlinearities

The equations of motion for a cantilevered beam are obtained from Hamilton's principle. This representation can include spanwise variations in mass and stiffness properties. The usual expression for strain energy — in local coordinates x spanwise, η chordwise, and ζ out-of-plane (see Fig. 3, taken from Hodges & Dowell [Ref. 13]) — in terms of engineering stresses and strains is,

$$(3-19) \qquad U = \frac{1}{2}\iiint\left(\sigma_{xx}\varepsilon_{xx} + \sigma_{\eta\eta}\varepsilon_{\eta\eta} + \sigma_{x\eta}\varepsilon_{x\eta} + \sigma_{x\zeta}\varepsilon_{x\zeta} + \sigma_{\eta\zeta}\varepsilon_{\eta\zeta}\right)dV$$

For an isotropic material, one usually sets $\varepsilon_{\eta\eta} = \varepsilon_{x\zeta} = \varepsilon_{\eta\zeta} = 0$ for beam theory. For an anisotropic material, however, it is usually necessary for consistency to instead set $\sigma_{\eta\eta} = \sigma_{x\zeta} = \sigma_{\eta\zeta} = 0$ — since the relations $\sigma_{\eta\eta} = E\varepsilon_{\eta\eta}$, etc. no longer hold — and then condense the stress strain relationships so that $\varepsilon_{\eta\eta}$, $\varepsilon_{x\zeta}$, and $\varepsilon_{\eta\zeta}$ are eliminated. For the current investigation, this more exact procedure of condensation has been avoided, although all the relations that will be derived are still valid, simply with condensed values (D_{11}^{*}, etc.) replacing non-condensed values (D_{11}, etc.) as in Ref. 74. After having eliminated $\varepsilon_{\eta\eta}$, $\varepsilon_{x\zeta}$, and $\varepsilon_{\eta\zeta}$, whether by condensation or by setting to zero, the first variation of the strain energy is given by,

$$(3-20) \qquad \delta U = \frac{1}{2}\iiint\left(\sigma_{xx}\delta\varepsilon_{xx} + \sigma_{x\eta}\delta\varepsilon_{x\eta}\right)dV$$

The strain components, after having been reduced to second order by the appropriate ordering scheme, are as given by Hodges & Dowell [Ref. 75],
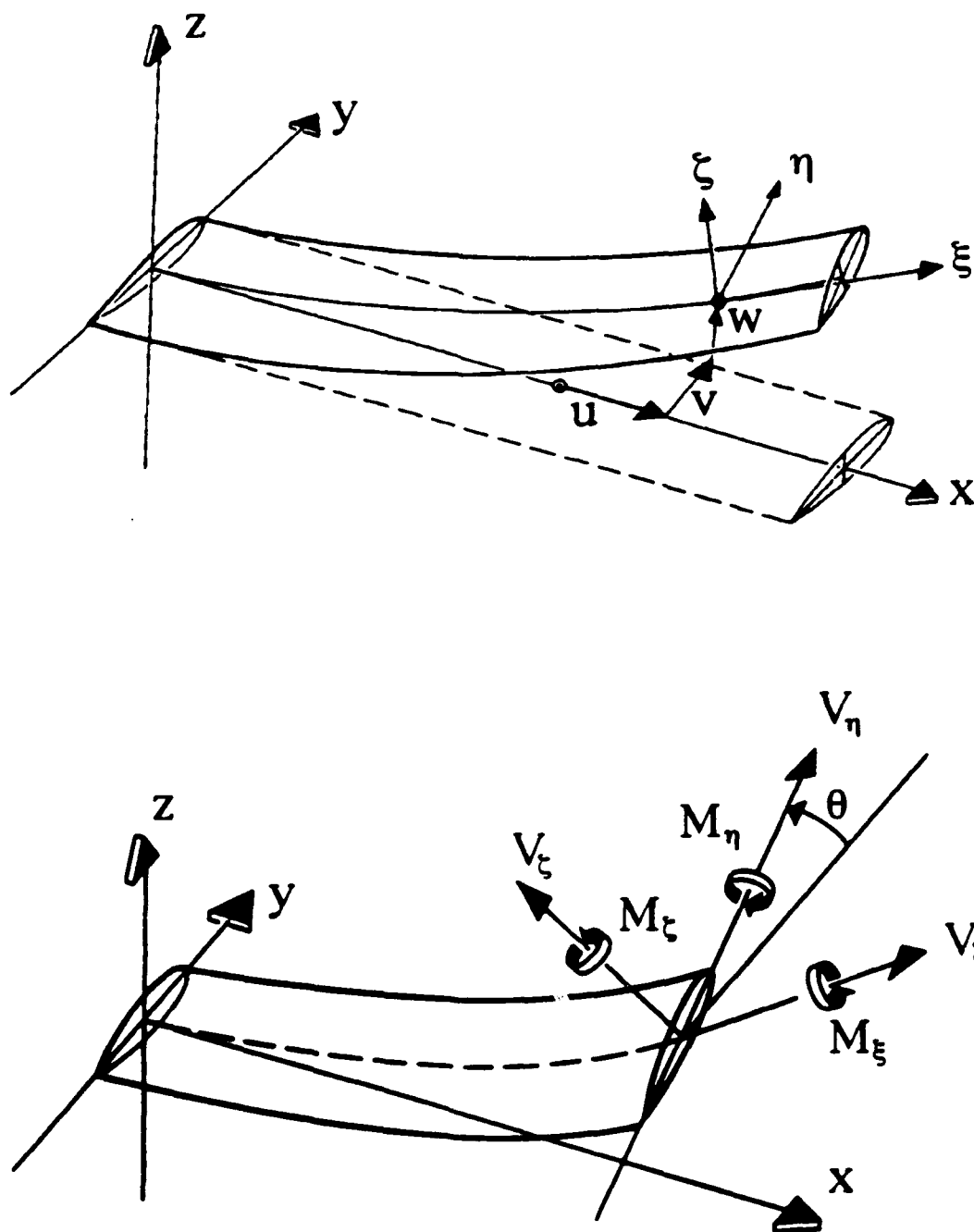
Fig. 3 Local coordinates and stress & moment resultants

$$(3-21) \qquad \varepsilon_{xx} = \frac{v'^2}{2} + \frac{w'^2}{2} + \left(\eta^2 + \zeta^2\right)\frac{\theta'^2}{2} - \lambda\theta'' - [\eta \cos\theta - \zeta \sin\theta]\, v''$$
$$- [\eta \sin\theta + \zeta \cos\theta]\, w''$$

$$(3-22) \qquad \varepsilon_{x\eta} = -\left(\zeta + \frac{\partial\lambda}{\partial\eta}\right)\theta'$$

For a thin plate the warping function can be approximated by, $\lambda = \zeta\sqrt{\eta^2 + \zeta^2} \approx \eta\zeta$, and therefore the strain components can be approximated by,

$$(3-23) \qquad \varepsilon_{xx} = \frac{v'^2}{2} + \frac{w'^2}{2} + \left(\eta^2 + \zeta^2\right)\frac{\theta'^2}{2} - \eta\zeta\theta'' - [\eta \cos\theta - \zeta \sin\theta]\, v''$$
$$- [\eta \sin\theta + \zeta \cos\theta]\, w''$$

$$(3-24) \qquad \varepsilon_{x\eta} = -2\zeta\theta'$$

Taking the variational of the strain components gives,

$$(3-25) \qquad \delta\varepsilon_{xx} = v'\delta v' + w'\delta w' + \left(\eta^2 + \zeta^2\right)\theta'\delta\theta' - \eta\zeta\delta\theta''$$
$$- [\eta \cos\theta - \zeta \sin\theta]\,(\delta v'' + w''\delta\theta)$$
$$- [\eta \sin\theta + \zeta \cos\theta]\,(\delta w'' - v''\delta\theta)$$

$$(3-26) \qquad \delta\varepsilon_{x\eta} = -2\zeta\delta\theta'$$

Substituting the expressions for the variational of the strain components, $\delta\varepsilon_{xx}$ and $\delta\varepsilon_{x\eta}$, into the equation for the strain energy variation, δU, we get an expression in terms of the stress and moment resultants (see Fig. 3),

23

$$(3\text{-}27) \qquad \delta U = \int_0^{\ell} \Big\{ V_x(v'\delta v' + w'\delta w') + (S_x + T_x)\delta\theta' + P_x\delta\theta''$$

$$+ [M_\zeta \cos\theta + M_\eta \sin\theta] \, (\delta v'' + w''\delta\theta)$$

$$+ [M_\zeta \sin\theta - M_\eta \cos\theta] \, (\delta w'' - v''\delta\theta) \Big\} \, dx$$

where the stress and moment resultants are defined as,

$$(3\text{-}28) \qquad V_x \equiv \iint \sigma_{xx} \, d\eta \, d\zeta = \int \sum_k \int \Big(Q_{11}^{(k)} \epsilon_{xx} + Q_{16}^{(k)} \epsilon_{x\eta} \Big) \, d\zeta \, d\eta$$

$$(3\text{-}29) \qquad S_x \equiv \iint -2\zeta\sigma_{x\eta} \, d\eta \, d\zeta = \int \sum_k \int -2\zeta \Big(Q_{16}^{(k)} \epsilon_{xx} + Q_{66}^{(k)} \epsilon_{x\eta} \Big) \, d\zeta \, d\eta$$

$$(3\text{-}30) \qquad T_x \equiv \iint \theta'\sigma_{xx}(\eta^2 + \zeta^2) \, d\eta \, d\zeta$$
$$= \int \sum_k \int (\eta^2 + \zeta^2)\theta' \Big(Q_{11}^{(k)} \epsilon_{xx} + Q_{16}^{(k)} \epsilon_{x\eta} \Big) \, d\zeta \, d\eta$$

$$(3\text{-}31) \qquad P_x \equiv \iint -\eta\zeta \, \sigma_{xx} \, d\eta \, d\zeta = \int \sum_k \int -\eta\zeta \Big(Q_{11}^{(k)} \epsilon_{xx} + Q_{16}^{(k)} \epsilon_{x\eta} \Big) \, d\zeta \, d\eta$$

$$(3\text{-}32) \qquad M_\eta \equiv \iint \zeta \, \sigma_{xx} \, d\eta \, d\zeta = \int \sum_k \int \zeta \Big(Q_{11}^{(k)} \epsilon_{xx} + Q_{16}^{(k)} \epsilon_{x\eta} \Big) \, d\zeta \, d\eta$$

$$(3\text{-}33) \qquad M_\zeta \equiv \iint -\eta \, \sigma_{xx} \, d\eta \, d\zeta = \int \sum_k \int -\eta \Big(Q_{11}^{(k)} \epsilon_{xx} + Q_{16}^{(k)} \epsilon_{x\eta} \Big) \, d\zeta \, d\eta$$

From equations (3-27), (3-29), and (3-30) one can see that S_x and T_x are respectively the shear and longitudinal stress contributions to the twisting moment resultant, i.e. $M_x = S_x + T_x$. Also, from equations (3-27) and (3-31) one can see that P_x is the torsional warping term in the strain energy equation. Finally, note that the axial stress and moment resultants use the subscript x instead of ξ as

24

in Fig. 3, because the deflections are only moderate and so they are almost equal, i.e. $V_x = V_\xi$, et cetera.

Substituting the previous nonlinear values for ϵ_{xx} and $\epsilon_{x\eta}$ and the evaluation of the integrals through the thickness by the appropriate flexural stiffness values, we get,

$$(3\text{-}34) \qquad V_x = EA\left\{\frac{v'^2}{2} + \frac{w'^2}{2} + k_A^2\frac{\theta'^2}{2} - e_A^*[w''\cos\theta - v''\sin\theta]\right\} - EB_3^{**}\theta'$$

$$(3\text{-}35) \qquad S_x = GJ_x\theta' + EI_{16}(w''\cos\theta - v''\sin\theta) - EB_3^{**}\left(\frac{v'^2}{2} + \frac{w'^2}{2}\right) - EB_4^{**}\frac{\theta'^2}{2}$$

$$(3\text{-}36) \qquad T_x = EAk_A^2\theta'\left(\frac{v'^2}{2} + \frac{w'^2}{2}\right) - EB_2^{**}\theta'(w''\cos\theta - v''\sin\theta) - EB_4^{**}\theta'^2$$

$$(3\text{-}37) \qquad P_x = EC_1\theta'' + EC_1^{**}(v''\cos\theta + w''\sin\theta)$$

$$(3\text{-}38) \qquad M_\eta = EI_\eta(v''\sin\theta - w''\cos\theta) - EI_{16}\theta' + EAe_A^*\left(\frac{v'^2}{2} + \frac{w'^2}{2}\right) + EB_2^{**}\frac{\theta'^2}{2}$$

$$(3\text{-}39) \qquad M_\zeta = EI_\zeta(v''\cos\theta + w''\sin\theta) + EC_1^{**}\theta''$$

where the stiffness properties are defined as,

$$(3\text{-}40) \qquad EA = \int\sum_k\int Q_{11}^{(k)}d\zeta\ d\eta = cA_{11}$$

$$(3\text{-}41) \qquad EAe_A^* = \int \sum_k \int \zeta Q_{11}^{(k)} \, d\zeta \, d\eta = cB_{11}$$

$$(3\text{-}42) \qquad EI_\eta = \int \sum_k \int \zeta^2 Q_{11}^{(k)} \, d\zeta \, d\eta = cD_{11}$$

$$(3\text{-}43) \qquad EI_\zeta = \int \sum_k \int \eta^2 Q_{11}^{(k)} \, d\zeta \, d\eta = \frac{c^3}{12} A_{11}$$

$$(3\text{-}44) \qquad EC_1^{**} = \int \sum_k \int \eta^2 \zeta Q_{11}^{(k)} \, d\zeta \, d\eta = \frac{c^3}{12} B_{11}$$

$$(3\text{-}45) \qquad EC_1 = \int \sum_k \int \eta^2 \zeta^2 Q_{11}^{(k)} \, d\zeta \, d\eta = \frac{c^3}{12} D_{11}$$

$$(3\text{-}46) \qquad EAk_A^2 = \int \sum_k \int (\eta^2 + \zeta^2) Q_{11}^{(k)} \, d\zeta \, d\eta = \frac{c^3}{12} A_{11} + cD_{11}$$

$$(3\text{-}47) \qquad EB_2^{**} = \int \sum_k \int \zeta(\eta^2 + \zeta^2) Q_{11}^{(k)} \, d\zeta \, d\eta = \frac{c^3}{12} B_{11} + c \sum_k \int \zeta^3 Q_{11}^{(k)} \, d\zeta$$

$$(3\text{-}48) \qquad GJ_x = \int \sum_k \int (2\zeta)^2 Q_{66}^{(k)} \, d\zeta \, d\eta = 4cD_{66}$$

$$(3\text{-}49) \qquad EI_{16} = \int \sum_k \int 2\zeta^2 Q_{16}^{(k)} \, d\zeta \, d\eta = 2cD_{16}$$

$$(3\text{-}50) \qquad EB_3^{**} = \int \sum_k \int 2\zeta Q_{16}^{(k)} \, d\zeta \, d\eta = 2cB_{16}$$

$$(3\text{-}51) \qquad EB_4^{**} = \int \sum_k \int 2\zeta(\eta^2 + \zeta^2)Q_{16}^{(k)}\, d\zeta\, d\eta$$

$$= 2\left[\frac{c^3}{12}B_{16} + c\sum_k \int \zeta^3 Q_{16}^{(k)}\, d\zeta\right]$$

It should be noted here the differences between the above formulation and that derived by Hodges & Dowell [Ref. 75]. Firstly, the derivation and values for EA, EAk_A^2, EI_η, EI_ζ, GJ_x, and EC_1 are the same. Secondly, the derivation and values for EAe_A^*, EB_2^{**}, and EC_1^{**} are not the same as for EAe_A, EB_2^*, and EC_1^* in Ref. 75 — but they are of the same form because the constants in the first set are derived from the asymmetry in stiffness in the through-the-thickness ζ-direction, in fashion similar to that in which the constants in the second set are derived from the asymmetry in geometry in the chordwise η-direction. Lastly, the EI_{16}, EB_3^{**}, and EB_4^{**} are new constants, not appearing in the Hodges & Dowell formulation, that result from the bending-shear coupling from the stacking sequence of an anisotropic material.

For a symmetric layup such as those considered in this investigation, EAe_A^*, EB_2^{**}, EC_1^{**}, EB_3^{**}, and EB_4^{**} are all zero. Also, there is no spanwise loading in the x-direction, so V_x is constant along the span, which indicates that $V_x=0$ from the root cantilevered conditions. Incorporating the easily-derived kinetic energy and generalized force terms, and applying integration by parts, yields,

$$(3\text{-}52) \qquad [M_\zeta \sin\theta - M_\eta \cos\theta]'' + m\ddot{w} = L_w$$

$$(3\text{-}53) \qquad [M_\zeta \cos\theta + M_\eta \sin\theta]'' + m\ddot{v} = L_v$$

27

$$(3\text{-}54) \qquad P_x'' - (S_x + T_x)' - v''[M_\zeta \sin\theta - M_\eta \cos\theta]$$

$$+ w''[M_\zeta \cos\theta + M_\eta \sin\theta] + i_x \ddot\theta = M_\theta$$

Substituting in the previously derived stress and moment resultants gives,

$$(3\text{-}55) \qquad \left[(EI_\eta \cos^2\theta + EI_\zeta \sin^2\theta)\, w'' + (EI_\zeta - EI_\eta)\cos\theta \sin\theta\, v'' \right.$$

$$\left. + EI_{16}\, \theta' \cos\theta \right]'' + m\ddot w = L_w$$

$$(3\text{-}56) \qquad \left[(EI_\zeta \cos^2\theta + EI_\eta \sin^2\theta)v'' + (EI_\zeta - EI_\eta)\cos\theta \sin\theta\, w'' \right.$$

$$\left. - EI_{16}\, \theta' \sin\theta \right]'' + m\ddot v = L_v$$

$$(3\text{-}57) \qquad \left[EAk_A^2\, \theta' \left(\frac{v'^2}{2} + \frac{w'^2}{2} \right) \right]' - (4cD_{66}\theta')' + (EC_1\theta'')''$$

$$+ (EI_\zeta - EI_\eta)[(w''^2 - v''^2)\cos\theta \sin\theta + v''w'' \cos2\theta]$$

$$- \left[EI_{16}(w''\cos\theta - v''\sin\theta) \right]'$$

$$- EI_{16}\theta'(v''\cos\theta + w''\sin\theta) + i_x \ddot\theta = M_\theta$$

Assume first a small amplitude vibration problem around a mean deflection. Now also assume an ordering scheme such that the mean out-of-plane deflection is moderate, while the torsion and fore-&-aft mean deflections are very small — i.e. that $w = \bar w + \tilde w$, $v = \bar v + \tilde v$, and $\theta = \bar\theta + \tilde\theta$, where the overbar indicates mean and the tilde indicates small deviation, and $\bar w/\ell = O(1)$, $\bar v/\ell = O(\epsilon)$, and $\bar\theta = O(\epsilon)$. With these assumptions, and dropping the tildes to indicate small deviation, the linearized form of the above equations is,

28

(3-58) $EI_\eta w^{iv} + EI_{16}\theta''' + m\ddot{w} = L_w$

(3-59) $EI_\zeta v^{iv} + (EI_\zeta - EI_\eta)(\bar{w}''\theta)'' + m\ddot{v} = L_v$

(3-60) $-GJ_x\theta'' + EC_1\theta^{iv} + (EI_\zeta - EI_\eta)(\bar{w}''^2\theta + v''\bar{w}'') - EI_{16}w''' + i_x\ddot{\theta} = M_\theta$

Note first that, again because V_x is zero, the first term from equation (3-57) — involving v'^2 and w'^2 — drops out. This is because, as can be seen from equation (3-34), if $V_x = 0$ and the layup is symmetric then the first term in equation (3-57) becomes $-EAk_A^4\theta'^3$, which drops out as a higher order term. Note also that equations (3-58) to (3-60) involve only the dynamic, small deviation terms of equations (3-55) to (3-57) — another three equations, similar in form but without the mass terms, also result so as to describe the mean deflections.

3.2 Aerodynamic Model

3.2.1 The ONERA Aerodynamic Model

The aerodynamic model used for this study was initially developed at *Office National d'Etudes et de Recherches Aérospatiales* by Tran & Petot [Ref. 49] and by Dat & Tran [Ref. 50]. This ONERA model is a semi-empirical, unsteady, nonlinear model which uses quasi-linear, small amplitude of oscillation, experimental data to predict aerodynamic forces on an oscillating airfoil which experiences dynamic stall. The model incorporates a single lag term operating on the linear part of the airfoil's static force curve, thus analogous to the Theodorsen function for linear theory, and a two lag term operating

on the nonlinear (i.e. stalling) portion of the airfoil's static force curve.

The ONERA model was later investigated by Peters [Ref. 55] who differentiated the roles of angle of attack due to pitching (θ) and angle of attack due to plunging ($\dot{h}/U$). The final form of the ONERA model used for this study incorporates all terms needed such that it fits the theoretical Theodorsen and Kussner coefficients within the linear domain of operation [Ref. 54],

$$(3\text{-}61) \qquad C_z = C_{z1} + C_{z2}$$

$$(3\text{-}62) \qquad C_{z1} = s_z \dot{\alpha} + k_{vz} \ddot{\theta} + C_{z\gamma}$$

$$(3\text{-}63) \qquad \dot{C}_{z\gamma} + \lambda_z C_{z\gamma} = \lambda_z \left[a_{oz}\alpha + \sigma_z \dot{\theta} \right] + \alpha_z \left[a_{oz}\dot{\alpha} + \sigma_z \ddot{\theta} \right]$$

$$(3\text{-}64) \qquad \ddot{C}_{z2} + a\dot{C}_{z2} + rC_{z2} = - r\Delta C_z \Big|_\alpha -e \frac{\partial \Delta C_z}{\partial \tau} \Big|_\alpha$$

where the effective angle of attack and time derivative are,

$$(3\text{-}65) \qquad \alpha = \theta - \dot{\bar{h}}$$

$$(3\text{-}66) \qquad (\dot{\ }) \equiv \frac{\partial(\ )}{\partial \tau} \ ; \ \tau \equiv \frac{Ut}{b}$$

and,

θ = instantaneous angle of attack

h = instantaneous deflection of 1/4-chord

$\bar{h} \equiv \frac{h}{b}$ = non-dimensional deflection

α = effective angle of attack

$C_{zs}(\alpha)$

extension of linear
static curve, $C_{z\ell}$

ΔC_z

non-linear
static curve

α
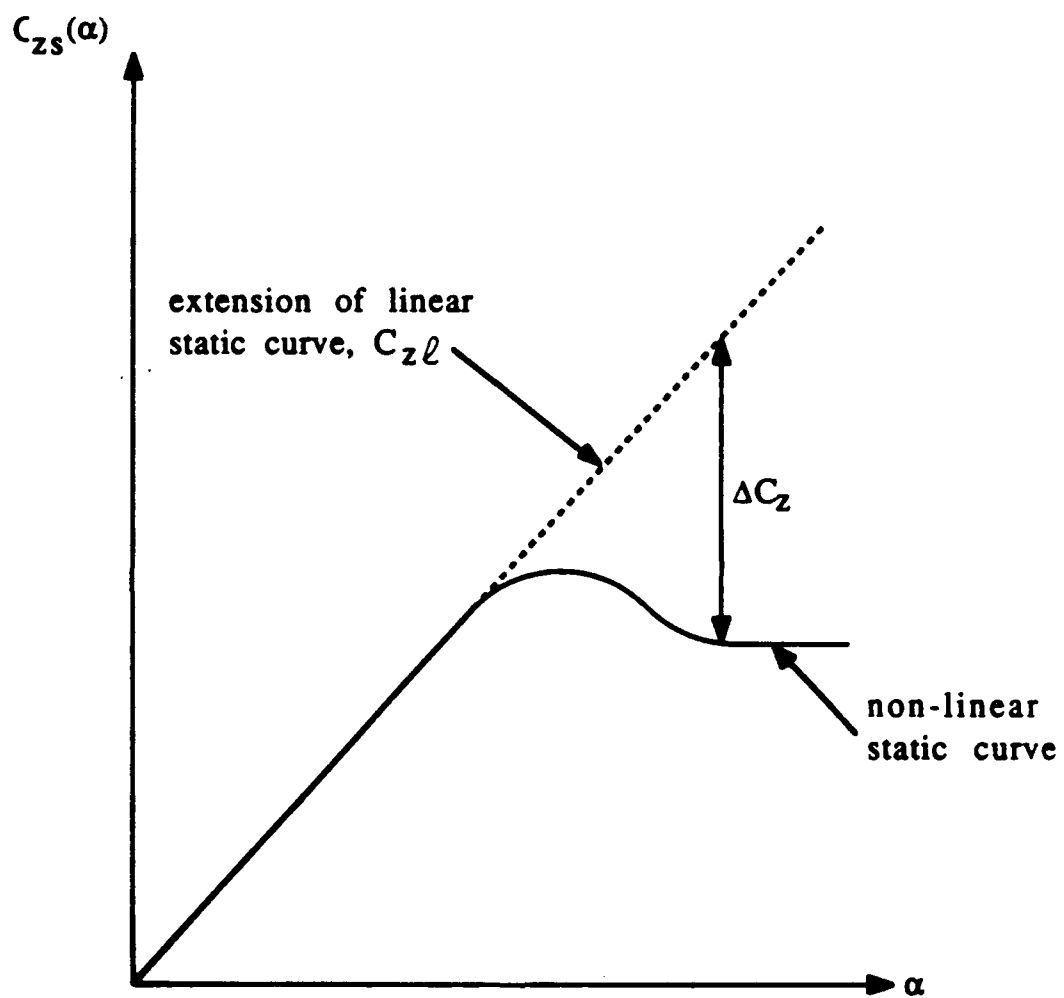
Fig. 4 Description of static curve

31

Note first that in equation (3-64) both forms of notation for the non-dimensional time derivative have been used, $\partial/\partial\tau$ and $(\overset{*}{})$ — this is only a matter of convenience of notation since it is awkward to use the $(\overset{*}{})$ notation over ΔC_z, and there is in fact no difference in the two time derivatives used on both sides of equation (3-64).

C_z represents any of the three relevant non-dimensional force coefficients: C_L, the coefficient of lift, or C_D, the coefficient of drag, or C_M, the moment coefficient. a_{oz} is the slope of the linear part of the static force curve, ΔC_z is the nonlinear deviation from the extended linear force curve, and s_z, k_{vz}, λ_z, σ_z, α_z, a, r, and e are the coefficients associated with the appropriate force coefficient, determined empirically by parameter identification. These force coefficients are listed in Appendix D.

Equations (3-62) and (3-63) describes that part of the force coefficient associated with the linear model C_{z1}, and are similar in form to the description of unsteady, linear theory with a first order lag for the Theodorsen function. $C_{z\gamma}$ is the linear circulatory contribution, while C_{z1} is the total linear contribution, also incorporating the apparent mass terms. Equation (3-64) describes that part of the force coefficient associated with the nonlinear model C_{z2}, and is dependent on the deviation of the actual static curve from the linear static curve, ΔC_z, as shown in Fig. 4. It also includes a second order lag for C_{z2}. Equation (3-61) combines these linear and nonlinear terms of the force coefficient into the total coefficient C_z.

For implementation of the ONERA aerodynamic model, it is necessary to describe the static aerodynamic force curves in terms of the linear domain, described by the linear slope a_{oz}, and the non-

32

linear domain, described by the deviation from the linear curve ΔC_z. The deviation ΔC_z is defined as positive for a decrease in the aerodynamic force, as shown in Fig. 4. The general description of the static aerodynamic force curve is then given by,

$$(3\text{-}67) \qquad C_{zs}(\alpha) = a_{0z}\alpha - \Delta C_z(\alpha)$$

where,

$$(3\text{-}68) \qquad a_{0z} = \frac{dC_{z\ell}}{d\alpha} = \text{linear aerodynamic force slope}$$

In general, the deviation ΔC_z can be described in any manner desired. In the current study, the deviation ΔC_z was described by simple straight line fits between discrete points (see Appendix C). More generally, the ΔC_z could be described by polynomials in several regions of the aerodynamic force curve. Polynomials of order J_i are used for ease of algebraic manipulations in the Fourier analysis, described later in Section 3.2.2.

The general formula for the deviation ΔC_z in the i-th region can then be expressed as,

$$(3\text{-}69) \qquad \Delta C_z(\alpha) = \sum_{j=0}^{J_i} a_{ij}(\alpha - \alpha_i)^j \quad ; \quad \alpha_i \le \alpha \le \alpha_{i+1}$$

where,

$$(3\text{-}70) \qquad a_{i0} = \Delta C_z(\alpha_i)$$

$$(3\text{-}71) \qquad \Delta C_z(\alpha=\alpha_1) = 0$$

33

Equation (3-68) ensures that the description of the aerodynamic force curve is continuous at the juncture of the describing domains. Equation (3-69) ensures that the deviation ΔC_z is identically zero in the linear region before stall. The description of the aerodynamic force coefficients used in the current study is more fully described in Appendix C.

3.2.2 Fourier Analysis of Nonlinear Aerodynamic Forcing Terms

For later use in the Harmonic Balance Method, it is necessary to be able to evaluate the lowest order frequency components of the nonlinear aerodynamic force coefficients when given a harmonic input. First, harmonic motion is assumed for the angle of attack and the non-dimensional, 1/4-chord deflection,

$$(3-72) \qquad \theta(\tau) = \theta_0 + \theta_s \sin(k\tau) + \theta_c \cos(k\tau)$$

$$(3-73) \qquad \bar{h}(\tau) = \bar{h}_0 + \bar{h}_s \sin(k\tau) + \bar{h}_c \cos(k\tau)$$

where,

$$k = \text{reduced frequency} = \frac{\omega b}{U}$$

$$\tau = \text{non-dimensional time} = \frac{Ut}{b}$$

The effective angle of attack, α, which combines both the instantaneous angle of attack and the angle of attack due to the velocity of the 1/4-chord deflection, is given by,

$$(3-74) \qquad \alpha(\tau) = \alpha_0 + \alpha_s \sin(k\tau) + \alpha_c \cos(k\tau)$$

where equation (3-65) gives,

$$(3-75) \qquad \alpha_o = \theta_o$$

$$(3-76) \qquad \alpha_s = \theta_s + k\bar{h}_c$$

$$(3-77) \qquad \alpha_c = \theta_c - k\bar{h}_s$$

Manipulations of the formulas are further simplified if the angle of attack is put in the form where it is purely sinusoidal,

$$(3-78) \qquad \alpha(\tau) = \alpha_o + \alpha_v \sin(k\tau + \xi) = \alpha_o + \alpha_v \sin\varphi$$

where,

$$(3-79) \qquad \alpha_v = \sqrt{\alpha_s^2 + \alpha_c^2}$$

$$(3-80) \qquad \xi = \tan^{-1}\frac{\alpha_c}{\alpha_s}$$

$$(3-81) \qquad \varphi = k\tau + \xi$$

Next, assume harmonic motion for ΔC_z as well,

$$(3-82) \qquad \Delta C_z(\tau) = \Delta C_{zo} + \Delta C_{zv1}\sin\varphi + \Delta C_{zv2}\cos 2\varphi + H.H.T$$

Note that equation (3-82) contains no $\cos\varphi$ and $\sin 2\varphi$ terms, i.e. no out-of-phase terms. This is because ΔC_z is a direct function of angle of attack α only — that is, $\Delta C_z = \Delta C_z(\alpha)$ and $\Delta C_z \neq \Delta C_z(\alpha, \tau)$ — without any lag terms, so the two are always in phase. If, however, a time dependence were added as well, for instance from a fixed-time stall

35

delay $\Delta\tau$, then there might be additional out-of-phase terms. Substituting equation (3-78) into equation (3-69) and carrying out the Fourier expansion yields,

$$(3-83) \qquad \Delta C_{zo} = \frac{1}{\pi} \sum_i \sum_{m=0}^{J_i} b_{im} I_{im}$$

where the b_{im}'s are the j-dependent terms, and the I_{im}'s are the j-independent terms, i.e. dependent or independent of the power of the describing polynomial. These terms are given by,

$$(3-84) \qquad b_{im} = \sum_{j=m}^{J_i} \binom{j}{m} a_{ij} \alpha_v^j \left(\frac{\alpha_o - \alpha_i}{\alpha_v} \right)^{j-m}$$

$$(3-85) \qquad I_{im} = \int_{\varphi_i}^{\varphi_{i+1}} \sin^m \varphi \, d\varphi = \left. \frac{-\sin^{m-1}\varphi \cos\varphi}{m} \right|_{\varphi_i}^{\varphi_{i+1}} + \frac{m-1}{m} I_{i,m-2}$$

where the limits of integration are given by transforming the region limits into the phase domain,

$$(3-86) \qquad \varphi_i = \begin{cases} \sin^{-1}\left(\dfrac{\alpha_i - \alpha_o}{\alpha_v} \right) \\[2mm] +\pi/2 \quad \text{if } \alpha_o + \alpha_v < \alpha_i \ \text{ i.e. } \alpha_i - \alpha_o > \alpha_v \\[2mm] -\pi/2 \quad \text{if } \alpha_o - \alpha_v > \alpha_i \ \text{ i.e. } \alpha_i - \alpha_o < -\alpha_v \end{cases}$$

and where the binomial coefficients are defined as,

$$(3-87) \qquad \binom{j}{m} = \frac{j!}{m! \, (j-m)!}$$

36

The first two values required for the recursive formula in equation (3-85) are given by,

$$(3\text{-}88) \qquad I_{i0} = \int_{\varphi_i}^{\varphi_{i+1}} d\varphi = \varphi_{i+1} - \varphi_i$$

$$(3\text{-}89) \qquad I_{i1} = \int_{\varphi_i}^{\varphi_{i+1}} \sin\varphi\, d\varphi = \cos\varphi_i - \cos\varphi_{i+1}$$

Similarly, using the same calculated b_{ij} and I_{ij} values, the first and second harmonic terms are given by,

$$(3\text{-}90) \qquad \Delta C_{zv1} = \frac{2}{\pi} \sum_i \sum_{m=0}^{J_i} b_{im} I_{i,m+1}$$

$$(3\text{-}91) \qquad \Delta C_{zv2} = \frac{2}{\pi} \sum_i \sum_{m=0}^{J_i} b_{im} (I_{i,m} - 2I_{i,m+2})$$

It is unnecessary to also carry out the full Fourier analysis for the time derivative of ΔC_z because of the mathematical identity that the Fourier expansion of the derivative of a function is equal to the derivative of the Fourier expansion. Hence, equation (3-82) gives,

$$(3\text{-}92) \qquad \frac{\partial \Delta C_z}{\partial \tau} = -k\Delta C_{zv1}\cos(k\tau) + 2k\Delta C_{zv2}\sin(2k\tau) + \text{H.H.T.}$$

where ΔC_{zv1} and ΔC_{zv2} are again given by equations (3-90) and (3-91). Simple examples for a force curve with only one and two break points are given in Appendix E.

37

3.2.3 Harmonic Balance Applied to ONERA Model

The harmonic balance method as applied to solving equation (3-64) begins by first assuming an infinite harmonic expansion for ΔC_z, as is given by equation (3-82). It is also assumed that C_{z2} can be expressed as an infinite harmonic series in the phase-shifted domain, i.e. in terms of $\varphi = k\tau + \xi$ instead of $k\tau$ — this is more convenient since the expression for ΔC_z is more simple in the phase-shifted domain,

$$(3\text{-}93) \qquad C_{z2} = C_{z2o} + \sum_{m} C_{z2sm}^{(\varphi)} \sin(m\varphi) + \sum_{n} C_{z2cn}^{(\varphi)} \cos(n\varphi)$$

As noted in previous studies [Ref. 53], it is most convenient to describe the nonlinear aerodynamic coefficients, a, r, & e, in terms of polynomial expressions in the lift deficit coefficient, ΔC_L. For a symmetric wing, the expression should be symmetric about $\Delta C_L = 0$, therefore a polynomial in even powers only, and it has generally been found that a simple parabolic is sufficient to describe these coefficients.

$$(3\text{-}94) \qquad a = a_0 + a_1 \Delta C_L^2$$

$$(3\text{-}95) \qquad r = \left[r_0 + r_1 \Delta C_L^2 \right]^2$$

$$(3\text{-}96) \qquad e = e_0 + e_1 \Delta C_L^2$$

Substituting the harmonic series for ΔC_L from equation (3-82) into the above equations yields expressions for a, r, & e in terms of the products of harmonic series, instead of constant coefficients

Substituting these harmonic expressions for a, r, & e, along with the harmonic series for ΔC_z and C_{z2}, from equations (3-82) and (3-93), into the unsteady, stalled ONERA equation (3-64), and then applying the trigonometric product identities,

$$\sin(x)\ \sin(y) = \tfrac{1}{2}(\cos(x-y) - \cos(x+y))$$

$$\cos(x)\ \cos(y) = \tfrac{1}{2}(\cos(x-y) + \cos(x+y))$$

$$\sin(x)\ \cos(y) = \tfrac{1}{2}(\sin(x-y) + \sin(x+y))$$

yields an infinite harmonic matrix equation for finding the harmonic components, C_{z2o}, $C_{z2s1}^{(\varphi)}$, $C_{z2c1}^{(\varphi)}$, etc., of the nonlinear contribution to the aerodynamic force, C_{z2}, given by equation (3-93). After allowing Mathematica™ to carry out the tedious algebra of multiplication of harmonic expansions and reduction of trigonometric products, the resulting equation is,

$$(3\text{-}97)\qquad \left[-\,k^2[I'] + \left[\mathcal{A}^{(r,a)}\right]\right]\{C_{z2}^{(\varphi)}\} = -\left[\mathcal{A}^{(r,e)}\right]\{\Delta C_z\}$$

where $-k^2[I']\{C_{z2}^{(\varphi)}\}$ comes from $\overset{*\ *}{C}_{z2}$; $\left[\mathcal{A}^{(r,a)}\right]\{C_{z2}^{(\varphi)}\}$ comes from $(a\overset{*}{C}_{z2}+rC_{z2})$; and $\left[\mathcal{A}^{(r,e)}\right]\{\Delta C_z\}$ from $(r\Delta C_z+r\partial(\Delta C_z)/\partial\tau)$. The elements of the nonlinear aerodynamic matrix $[\mathcal{A}]$ truncated to two harmonics (i.e. 5x5) are,

$$(3\text{-}98)\qquad \mathcal{A}_{11}^{(r,a)} = R\Big[(1 + \beta_2(\Delta_1^2 + \Delta_2^2) + \beta_2^2(2\Delta_0^2\Delta_1^2 + \tfrac{3}{8}\Delta_1^4 - 3\Delta_0\Delta_1^2\Delta_2$$

$$+ 2\Delta_0^2\Delta_2^2 + \tfrac{3}{2}\Delta_1^2\Delta_2^2 + \tfrac{3}{8}\Delta_2^4)\Big]$$

$$(3\text{-}99) \quad A_{12}^{(r,a)} = R\left[\beta_2(2\Delta_0\Delta_1 - \Delta_1\Delta_2) + \beta_2^2(\tfrac{3}{2}\Delta_0\Delta_1^3 - 2\Delta_0^2\Delta_1\Delta_2 - \Delta_1^3\Delta_2\right.$$
$$\left. + 3\Delta_0\Delta_1\Delta_2^2 - \tfrac{3}{4}\Delta_1\Delta_2^3)\right]$$

$$(3\text{-}100) \quad A_{13}^{(r,a)} = -kA\beta_1(\Delta_0\Delta_1 - \tfrac{1}{2}\Delta_1\Delta_2)$$

$$(3\text{-}101) \quad A_{14}^{(r,a)} = 2kA\beta_1(\tfrac{-1}{4}\Delta_1^2 + \Delta_0\Delta_2)$$

$$(3\text{-}102) \quad A_{15}^{(r,a)} = R\left[\beta_2(\tfrac{-1}{2}\Delta_1^2 + 2\Delta_0\Delta_2) + \beta_2^2(-\Delta_0^2\Delta_1^2 - \tfrac{1}{4}\Delta_1^4 + 3\Delta_0\Delta_1^2\Delta_2\right.$$
$$\left. - \tfrac{9}{8}\Delta_1^2\Delta_2^2 + \tfrac{3}{2}\Delta_0\Delta_2^3)\right]$$

$$(3\text{-}103) \quad A_{21}^{(r,a)} = R\left[\beta_2(4\Delta_0\Delta_1 - 2\Delta_1\Delta_2) + \beta_2^2(3\Delta_0\Delta_1^3 - 4\Delta_0^2\Delta_1\Delta_2\right.$$
$$\left. - 2\Delta_1^3\Delta_2 + 6\Delta_0\Delta_1\Delta_2^2 - \tfrac{3}{2}\Delta_1\Delta_2^3)\right]$$

$$(3\text{-}104) \quad A_{22}^{(r,a)} = R\left[1 + \beta_2(\tfrac{3}{2}\Delta_1^2 - 2\Delta_0\Delta_2 + \Delta_2^2) + \beta_2^2(3\Delta_0^2\Delta_1^2 + \tfrac{5}{8}\Delta_1^4\right.$$
$$\left. - 6\Delta_0\Delta_1^2\Delta_2 + 2\Delta_0^2\Delta_2^2 + \tfrac{21}{8}\Delta_1^2\Delta_2^2 - \tfrac{3}{2}\Delta_0\Delta_2^3 + \tfrac{3}{8}\Delta_2^4)\right]$$

$$(3\text{-}105) \quad A_{23}^{(r,a)} = -kA\left[1 + \beta_1(\tfrac{3}{4}\Delta_1^2 - \Delta_0\Delta_2 + \tfrac{1}{2}\Delta_2^2)\right]$$

$$(3\text{-}106) \quad A_{24}^{(r,a)} = 2kA\beta_1(-\Delta_0\Delta_1 + \Delta_1\Delta_2)$$

$$(3\text{-}107) \quad A_{25}^{(r,a)} = R\Big[\beta_2(-2\Delta_0\Delta_1 + 2\Delta_1\Delta_2) + \beta_2^2(-2\Delta_0\Delta_1^3 + 4\Delta_0^2\Delta_1\Delta_2$$
$$+ \tfrac{7}{4}\Delta_1^3\Delta_2 - \tfrac{9}{2}\Delta_0\Delta_1\Delta_2^2 + \tfrac{3}{2}\Delta_1\Delta_2^3)\Big]$$

$$(3\text{-}108) \quad A_{31}^{(r,a)} = 0$$

$$(3\text{-}109) \quad A_{32}^{(r,a)} = kA\Big[1 + \beta_1(\tfrac{1}{4}\Delta_1^2 + \Delta_0\Delta_2 + \tfrac{1}{2}\Delta_2^2)\Big]$$

$$(3\text{-}110) \quad A_{33}^{(r,a)} = R\Big[1 + \beta_2(\tfrac{1}{2}\Delta_1^2 + 2\Delta_0\Delta_2 + \Delta_2^2) + \beta_2^2(\Delta_0^2\Delta_1^2 + \tfrac{1}{8}\Delta_1^4 + 2\Delta_0^2\Delta_2^2$$
$$+ \tfrac{3}{8}\Delta_1^2\Delta_2^2 + \tfrac{3}{2}\Delta_0\Delta_2^3 + \tfrac{3}{8}\Delta_2^4)\Big]$$

$$(3\text{-}111) \quad A_{34}^{(r,a)} = R\Big[2\beta_2\Delta_0\Delta_1 + \beta_2^2(\Delta_0\Delta_1^3 - \tfrac{1}{4}\Delta_1^3\Delta_2 + \tfrac{3}{2}\Delta_0\Delta_1\Delta_2^2)\Big]$$

$$(3\text{-}112) \quad A_{35}^{(r,a)} = -2kA\beta_1\Delta_0\Delta_1$$

$$(3\text{-}113) \quad A_{41}^{(r,a)} = 0$$

$$(3\text{-}114) \quad A_{42}^{(r,a)} = kA\beta_1\Delta_0\Delta_1$$

$$(3\text{-}115) \quad A_{43}^{(r,a)} = R\Big[2\beta_2\Delta_0\Delta_1 + \beta_2^2(\Delta_0\Delta_1^3 - \tfrac{1}{4}\Delta_1^3\Delta_2 + \tfrac{3}{2}\Delta_0\Delta_1\Delta_2^2)\Big]$$

$$(3\text{-}116) \quad A_{44}^{(r,a)} = R\Big[1 + \beta_2(\Delta_1^2 + \tfrac{1}{2}\Delta_2^2) + \beta_2^2(2\Delta_0^2\Delta_1^2 + \tfrac{5}{16}\Delta_1^4 - \tfrac{3}{2}\Delta_0\Delta_1^2\Delta_2$$
$$+ \Delta_0^2\Delta_2^2 + \tfrac{3}{4}\Delta_1^2\Delta_2^2 + \tfrac{1}{8}\Delta_2^4)\Big]$$

(3-117) $\quad \beta_{45}^{(r,a)} = -2kA\left[1 + \beta_1(\tfrac{1}{2}\Delta_1^2 + \tfrac{1}{4}\Delta_2^2)\right]$

(3-118) $\quad \beta_{51}^{(r,a)} = R\left[\beta_2(\Delta_1^2 + 4\Delta_0\Delta_2) + \beta_2^2(-4\Delta_0^2\Delta_1^2 - \tfrac{1}{2}\Delta_1^4 + 6\Delta_0\Delta_1^2\Delta_2 \right.$

$$\left. - \tfrac{9}{4}\Delta_1^2\Delta_2^2 + 3\Delta_0\Delta_2^3)\right]$$

(3-119) $\quad \beta_{52}^{(r,a)} = R\left[\beta_2(-2\Delta_0\Delta_1 + 2\Delta_1\Delta_2) + \beta_2^2(-2\Delta_0\Delta_1^3 + 4\Delta_0^2\Delta_1\Delta_2 \right.$

$$\left. + \tfrac{7}{4}\Delta_1^3\Delta_2 - \tfrac{9}{2}\Delta_0\Delta_1\Delta_2^2 + \tfrac{3}{2}\Delta_1\Delta_2^3)\right]$$

(3-120) $\quad \beta_{53}^{(r,a)} = -kA\beta_1(\Delta_0\Delta_1 + \Delta_1\Delta_2)$

(3-121) $\quad \beta_{54}^{(r,a)} = 2kA\left[1 + \beta_1(\tfrac{1}{2}\Delta_1^2 + \tfrac{3}{4}\Delta_2^2)\right]$

(3-122) $\quad \beta_{55}^{(r,a)} = R\left[1 + \beta_2(\Delta_1^2 + \tfrac{3}{2}\Delta_2^2) + \beta_2^2(2\Delta_0^2\Delta_1^2 + \tfrac{7}{16}\Delta_1^4 - \tfrac{9}{2}\Delta_0\Delta_1^2\Delta_2 \right.$

$$\left. + 3\Delta_0^2\Delta_2^2 + \tfrac{9}{4}\Delta_1^2\Delta_2^2 + \tfrac{5}{8}\Delta_2^4)\right]$$

where the intermediate variables are given by,

(3-123) $\quad A = (a_0 + a_1\Delta_0^2) \qquad ; \beta_1 = a_1/A$

(3-124) $\quad R = \left[r_0 + r_1\Delta_0^2\right]^2 \qquad ; \beta_2 = r_1/\sqrt{R}$

(3-125) $\quad E = (e_0 + e_1\Delta_0^2) \qquad ; \beta_3 = e_1/E$

and the shorthand notation $\Delta_0 = \Delta C_{L0}$, $\Delta_1 = \Delta C_{Lv1}$, and $\Delta_2 = \Delta C_{Lv2}$ has been used, as derived in Section 3.2.2, and given by equation (3-82). The magnitudes of the nonlinear coupling terms, β_1, β_2, & β_3, go like

42

a_1/a_0, r_1/r_0, & e_1/e_0 because the flutter analyses are usually only in light stall, i.e. when Δ_0 is relatively small. Therefore, from Appendix D, the nonlinear coupling terms are of the order of $\beta_1 \approx 2$, $\beta_2 \approx 10$, and $\beta_3 \approx \infty$.

Note that A, R, & E are the values of a, r, & e based only on the mean value of ΔC_{L0} — i.e. they are the simplified values used in Refs. 94 and 95, and that same analysis can be reproduced by simply setting β_1, β_2, & β_3 to zero. Moreover, for small amplitudes of oscillation, i.e. on the flutter boundary, Δ_1 & Δ_2 are negligible relative to Δ_0, and again the analysis can be much simplified by setting β_1, β_2, & β_3 to zero.

The aerodynamic matrix $\left[A^{(r,e)} \right]$ governing the right hand side of equation (3-97) is identical to the aerodynamic matrix $\left[A^{(r,a)} \right]$ just derived, only with all a values substituted by e values. This would not be the case unless the nonlinear coefficients a & e both had the same parabolic form, as in equations (3-94) and (3-96). Hence,

$$(3\text{-}126) \quad \left[A^{(r,e)} \right] = \left\{ \left[A^{(r,a)} \right] : a_0 \rightarrow e_0, \, a_1 \rightarrow e_1 \right\}$$

$$= \left\{ \left[A^{(r,a)} \right] : A \rightarrow E, \, \beta_1 \rightarrow \beta_3 \right\}$$

The [I'] matrix is like an identity matrix, but with squares of the ascending integers taking up every two diagonal elements (one each for the sine and cosine components). This matrix results from taking the second derivative of the harmonics with respect to time. Truncated to the second harmonic, it looks like,

$$(3\text{-}127) \quad [I'] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

The $\{C_{z2}^{(\varphi)}\}$ and $\{\Delta C_z\}$ column vectors of equation (3-97) are made up of the harmonic elements of C_{z2} and ΔC_z from equations (3-82) and (3-93)

$$(3\text{-}128) \quad \{C_{z2}^{(\varphi)}\} = \lfloor C_{z20} \; C_{z2s1}^{(\varphi)} \; C_{z2c1}^{(\varphi)} \; C_{z2s2}^{(\varphi)} \; C_{z2c2}^{(\varphi)} \; ... \rfloor^T$$

$$(3\text{-}129) \quad \{\Delta C_z\} = \lfloor \Delta C_{zo} \; \Delta C_{zv1} \; 0 \quad 0 \quad \Delta C_{zv2} \; ... \rfloor^T$$

Matrix equations (3-97) is solved to find the phase-shifted harmonic components of C_{z2}, and these are then converted into the real-time domain,

$$(3\text{-}130) \quad C_{z2s1} = C_{z2s1}^{(\varphi)} \cos\xi - C_{z2c1}^{(\varphi)} \sin\xi$$

$$(3\text{-}131) \quad C_{z2c1} = C_{z2c1}^{(\varphi)} \cos\xi + C_{z2s1}^{(\varphi)} \sin\xi$$

$$(3\text{-}132) \quad C_{z2s2} = C_{z2s2}^{(\varphi)} \cos2\xi - C_{z2c2}^{(\varphi)} \sin2\xi$$

$$(3\text{-}133) \quad C_{z2c2} = C_{z2c2}^{(\varphi)} \cos2\xi + C_{z2s2}^{(\varphi)} \sin2\xi$$

3.3 Algebraic Reduction by Modal Analysis

3.3.1 Rayleigh-Ritz Analysis

The direct Rayleigh-Ritz energy method is a relatively simple, straightforward approximation for the plate deflections, as required for the static deflection, free vibration, and flutter analyses in this

44

study. The Rayleigh-Ritz method also has the advantage of showing the effect of the individual variables on the solution more clearly than other more accurate methods, such as finite element analysis. The "wing" is idealized by a rectangular, cantilevered, graphite/epoxy flat plate of uniform thickness, with styrofoam fairings covering the entire chord but only part of the entire span.

The Rayleigh-Ritz analysis begins by assuming a deflection shape for the structure. If only out-of-plane deflections, w, and fore-&-aft deflections, v, are allowed, the deflection equations, written in generalized coordinates, are,

$$(3\text{-}134) \qquad w = \sum_{i=1}^{n} \gamma_i^w(x,y) q_i(t)$$

$$(3\text{-}135) \qquad v = \sum_{i=1}^{n} \gamma_i^v(x,z) q_i(t)$$

where $\gamma_i^w(x,y)$ and $\gamma_i^v(x,y)$ together are the non-dimensional deflection, or mode shape, of the i-th mode; $q_i(t)$ is the generalized displacement, or modal amplitude, of the i-th mode; and n is the number of mode shapes.

For simplicity, it is further assumed that the mode shapes are separable in the chordwise, spanwise, and through-the-thickness directions — x, y, and z — namely that the mode shape can be written in the form,

$$(3\text{-}136) \qquad \gamma_i^w(x,y) = \phi_i^w(x) \psi_i^w(y)$$

$$(3\text{-}137) \qquad \gamma_i^v(x,z) = \phi_i^v(x)\psi_i^v(z)$$

The nonlinear equations of motion are thus transformed from equations that are differential in both space and time, to equations that are algebraic in space and differential in time. These equation — relating the modal amplitudes to the modal forces [Ref. 70] — are,

$$(3\text{-}138) \qquad \sum_{j=1}^{n} M_{ij}\ddot{q}_j + \sum_{j=1}^{n} K_{ij}q_j = Q_i \quad i=1,\dots,n$$

or, in matrix form,

$$(3\text{-}139) \qquad [M]\{\ddot{q}\} + [K]\{q\} = \{Q\}$$

The mass and stiffness matrices are comprised of contributions from the graphite/epoxy flat plate, the styrofoam fairings, and effects of geometric nonlinearities, while the aerodynamic forces contribute to the modal forces,

$$(3\text{-}140) \qquad M_{ij} = M_{ij}^{fp} + M_{ij}^{sty} + M_{ij}^{geo}$$

$$(3\text{-}141) \qquad K_{ij} = K_{ij}^{fp} + K_{ij}^{sty} + K_{ij}^{geo}$$

The derivations of these mass and stiffness contributions and of the modal forces are described in Section 3.3.3.

3.3.2 Selection of Rayleigh-Ritz Modes

To sufficiently describe the deflection of the wing in the static bending, free vibration and flutter tests, beam out-of-plane bending

46

modes, beam torsion modes, chordwise bending modes, and beam fore-&-aft bending modes were chosen. Previous studies used simplified, sinusoidal torsional mode shapes [Ref. 92] that did not meet the cantilevered root conditions, but with a torsional stiffness correction which accounted for the effect of root warping stiffness [Ref. 70]. Vibrations tests, where the modal amplitudes were very small and the modal forces identically zero, showed that this torsional stiffness correction sufficed to accurately predict the natural frequencies and modes of vibration of the wings.

However, static bending tests and low speed, steady deflection, wind tunnel tests conducted in this study, where the modal amplitudes and modal forces were no longer insignificant, showed that the use of mode shapes which did not meet the cantilevered root condition adversely affected the Rayleigh-Ritz prediction of modal deflections. Therefore, the more complex torsional modes, with similar spanwise form as the beam bending modes, were used instead of the simplified sinusoidal mode shapes. The selected mode shapes are listed in Table 1.

The parameters of the beam torsion modes are derived from the definition of β and the relationship between f_j and g_j,

$$(3-142) \qquad \beta = \frac{D_{11}c^2}{48D_{66}\ell^2}$$

$$(3-143) \qquad f^2 = g^2 + \frac{1}{\beta}$$

47

<u>Out-of-plane bending:</u> mode # = i = j = 1 to n_b

$$\phi_i^w(x) = \cosh\left(\varepsilon_j\frac{x}{\ell}\right) - \cos\left(\varepsilon_j\frac{x}{\ell}\right) - \alpha_j\left[\sinh\left(\varepsilon_j\frac{x}{\ell}\right) - \sin\left(\varepsilon_j\frac{x}{\ell}\right)\right]$$

$$\psi_i^w(y) = 1$$

$$\gamma_i^v(x,z) = 0$$

$$\varepsilon_j = \rho_j\pi \ ; \ \alpha_j = \frac{\sinh \varepsilon_j - \sin \varepsilon_j}{\cosh \varepsilon_j + \cos \varepsilon_j}$$

$$\rho_j = \ 0.596864162695, \quad 1.494175614274, \quad 2.500246946168,$$
$$3.499989319849, \quad 4.500000461516, \quad 5.5, \quad 6.5, \quad \ldots$$

<u>Torsion:</u> mode # = i = n_b+j ; j = 1 to n_t

$$\phi_i^w(x) = B_{j1}\cos\left(g_j\frac{x}{\ell}\right) + B_{j2}\sin\left(g_j\frac{x}{\ell}\right) + B_{j3}\cosh\left(f_j\frac{x}{\ell}\right) + B_{j4}\sinh\left(f_j\frac{x}{\ell}\right)$$

$$\psi_i^w(y) = \frac{y}{c}$$

$$\gamma_i^v(x,z) = 0$$

<u>Chordwise bending:</u> mode # = i = n_b+n_t+j ; j = 1 to n_c

$$\phi_1(x) = \frac{x}{\ell}\left(1 - \frac{x}{\ell}\right) \quad ; \ \psi_1(y) = \frac{4y^2}{c^2} - \frac{1}{3} \quad ; \ \gamma_i^v(x,z) = 0$$

$$\phi_2(x) = \frac{x^2}{\ell^2} - 1 \quad ; \ \psi_2(y) = \frac{4y^2}{c^2} - 1 \quad ; \ \gamma_i^v(x,z) = 0$$

<u>Fore-&-aft:</u> mode # = i = n_b+n_t+n_c+j ; j = 1 to n_f

$$\gamma_i^w(z) = 0$$

$$\phi_i^v(x) = \cosh\left(\varepsilon_j\frac{x}{\ell}\right) - \cos\left(\varepsilon_j\frac{x}{\ell}\right) - \alpha_j\left[\sinh\left(\varepsilon_j\frac{x}{\ell}\right) - \sin\left(\varepsilon_j\frac{x}{\ell}\right)\right]$$

$$\psi_i^v(z) = 1$$

ε_j and α_j same as above

Table 1. Assumed mode shapes

and by solving the coupled equations which ensure that the mode shape meets the plate boundary conditions at the root and tip. For the assumed torsional mode shapes, the equations that describe the boundary condition that must be met can be written in matrix form as [Ref. 69],

$$(3\text{-}144) \quad \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & g & 0 & f \\ -g^2\cos g & -g^2\sin g & f^2\cosh f & f^2\sinh f \\ gf^2\sin g & gf^2\cos g & g^2f\sinh f & g^2f\cosh f \end{bmatrix} \begin{bmatrix} B_{j1} \\ B_{j2} \\ B_{j3} \\ B_{j4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The first two lines of the matrix equation (3-144) ensure that the deflection and slope at the plate root are zero. The last two lines of the matrix equation ensure that the internal forces at the plate tip are also zero.

Since f and g are related through equation (3-143), the non-trivial solution to the eigenvalue problem is found by setting the determinant of the matrix in equation (3-144) to zero. The values for f and g can be found by a simple Newton solver scheme. Once the f and g values are found, the B_{ij} coefficients are determined through the following matrix equation,

$$(3\text{-}145) \quad \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & g & 0 & f \\ -g^2\cos g & -g^2\sin g & f^2\cosh f & f^2\sinh f \\ \cos g & \sin g & \cosh f & \sinh f \end{bmatrix} \begin{bmatrix} B_{j1} \\ B_{j2} \\ B_{j3} \\ B_{j4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The fourth line in equation (3-145), which normalizes the modal tip deflection to one, replaces the fourth line of equation

(3-144), which becomes redundant when f and g are solved so as to make the matrix singular.

It is clear from equation (3-145) that B_{i1} and B_{i3} are equal in magnitude but opposite in sign. It is also generally found that f is order of magnitude 10, so that the cosh(f) and sinh(f) terms dominate the third line of the matrix equation, making B_{i3} and B_{i4} opposite in sign and almost equal in magnitude. It is important to note for purposes of calculating the tip deflection that B_{i3} and B_{i4} are not exactly equal in magnitude, since this difference is magnified exponentially by the cosh and sinh terms near $x/\ell=1$. Values of β, f, g, and B_{ij} for the layups used in this study are listed in Appendix B.

3.3.3 Mass & Stiffness Matrices and Modal Forces

For the flat plate, the symmetric mass coefficients, M_{ij}^{fp}, and the symmetric stiffness coefficients, K_{ij}^{fp}, are defined as,

$$(3-146) \quad M_{ij}^{fp} = \iiint \rho\, \gamma_i^w \gamma_j^w\, dxdydz \quad \text{for } i,j \leq n_b+n_t+n_c$$

$$(3-147) \quad M_{ij}^{fp} = \iiint \rho\, \gamma_i^v \gamma_j^v\, dxdydz \quad \text{for } n_b+n_t+n_c < i,j$$

$$(3-148) \quad K_{ij}^{fp} = \iint \{ D_{11}\gamma_{i,xx}^w\gamma_{j,xx}^w + D_{22}\gamma_{i,yy}^w\gamma_{j,yy}^w + 4D_{66}\gamma_{i,xy}^w\gamma_{j,xy}^w$$

$$+ D_{12}[\gamma_{i,xx}^w\gamma_{j,yy}^w + \gamma_{i,yy}^w\gamma_{j,xx}^w]$$

$$+ 2D_{16}[\gamma_{i,xx}^w\gamma_{j,xy}^w + \gamma_{i,xy}^w\gamma_{j,xx}^w]$$

$$+ 2D_{26}[\gamma_{i,yy}^w\gamma_{j,xy}^w + \gamma_{i,xy}^w\gamma_{j,yy}^w]\}dxdy$$

$$\text{for } i,j \leq n_b+n_t+n_c$$

$$(3\text{-}149) \quad K_{ij}^{fp} = \frac{c^2}{12} \iint \{ A_{11}\gamma_{i,xx}^v \gamma_{j,xx}^v + A_{22}\gamma_{i,zz}^v \gamma_{j,zz}^v + 4A_{66}\gamma_{i,xz}^v \gamma_{j,xz}^v$$

$$+ A_{12}[\gamma_{i,xx}^v \gamma_{j,zz}^v + \gamma_{i,zz}^v \gamma_{j,xx}^v] + 2A_{16}[\gamma_{i,xx}^v \gamma_{j,xz}^v + \gamma_{i,xz}^v \gamma_{j,xx}^v]$$

$$+ 2A_{26}[\gamma_{i,zz}^v \gamma_{j,xz}^v + \gamma_{i,xz}^v \gamma_{j,zz}^v] \} dx dz \quad \text{for } n_b + n_t + n_c < i,j$$

where ρ is the density and the subscripts following the commas denote partial differentiation with respect to the spatial coordinates, x, y, and z. A 12-point Gaussian quadrature scheme was used to evaluate the above integrals.

The styrofoam contributions to the mass matrix can be calculated in the same form as equations (3-146) and (3-147) using the known thickness of the styrofoam and the chosen mode shapes from Table 1.

$$(3\text{-}150) \quad M_{ij}^{sty} = \rho_s c (0.685 t_{max} - t_{fp}) \int_{x_{sty}}^{\ell} \phi_i \phi_j dx$$

$$\text{for } i,j \leq n_b \text{ or } n_b + n_t + n_c < i,j$$

$$(3\text{-}151) \quad M_{ij}^{sty} = \rho_s \frac{c}{12} (0.506 t_{max} - t_{fp}) \int_{x_{sty}}^{\ell} \phi_i \phi_j dx$$

$$\text{for } n_b < i,j \leq n_b + n_t$$

$$(3\text{-}152) \quad M_{ij}^{sty} = -0.0545 \rho_s c t_{max} \int_{x_{sty}}^{\ell} \phi_i \phi_j dx$$

$$\text{for } i \leq n_b; \; n_b < j \leq n_b + n_t$$

51

The calculations of the styrofoam contributions to the mass matrix from the chordwise bending modes are more cumbersome because of the complicated chordwise variation of the mode shape. Therefore, for those components of the mass matrix involving the chordwise bending modes, the styrofoam thickness is assumed to be uniformly half the maximum thickness, t_{max}, as might be suggested by equation (3-151).

In the same manner, the contributions of the styrofoam to the stiffness matrix can be calculated, giving,

$$(3-153) \quad K_{ij}^{sty} = 2cQ_{11}^{sty} \frac{\left(\frac{.779t_{max}}{2}\right)^3 - \left(\frac{t_{fp}}{2}\right)^3}{3} \int_{x_{sty}}^{\ell} \phi_{i,xx}\phi_{j,xx}dx$$

for $i,j \leq n_b$ or $n_b + n_t + n_c < i,j$

$$(3-154) \quad K_{ij}^{sty} = -.01585cQ_{11}^{sty}t_{max}^3 \int_{x_{sty}}^{\ell} \phi_{i,xx}\phi_{j,xx}dx$$

for $i \leq n_b$; $n_b < j \leq n_b + n_t$

$$(3-155) \quad K_{ij}^{sty} = \frac{1}{12}cQ_{11}^{sty} \frac{\left(\frac{.824t_{max}}{2}\right)^3 - \left(\frac{t_{fp}}{2}\right)^3}{3} \int_{x_{sty}}^{\ell} \phi_{i,xx}\phi_{j,xx}dx +$$

$$+ 8cQ_{66}^{sty} \frac{\left(\frac{.779t_{max}}{2}\right)^3 - \left(\frac{t_{fp}}{2}\right)^3}{3} \int_{x_{sty}}^{\ell} \frac{\phi_{i,x}\phi_{j,x}}{c \quad c}dx$$

for $n_b < i,j \leq n_b + n_t$

52

where Q_{11}^{sty} and Q_{66}^{sty} are the styrofoam engineering constants, defined in the same manner as for the graphite/epoxy in equations (3-1) and (3-4), as listed in Appendix A.

Again, the calculations involving the chordwise bending mode are quite cumbersome, so for these purposes the styrofoam is assumed to be uniformly 80% its maximum thickness, as might be suggested by equations (3-153) and (3-155).

The contributions from the geometric nonlinearities can be added by applying the same Rayleigh-Ritz method to the nonlinear equations of motion derived in Section 3.1.2. There are no nonlinear contributions to the mass matrix, $M_{ij}^{geo}=0$. The nonlinear contributions to the stiffness matrix, similar to those derived by Boyd [Ref. 78], are,

$$(3\text{-}156) \quad K_{ij}^{geo} = \begin{cases} \displaystyle\sum_{m=1}^{n_b} \sum_{n=1}^{n_b} R_{mnij}q_m q_n & \text{for } n_b<i,j\leq n_b+n_t \\ \displaystyle\sum_{m=1}^{n_b} H_{mij}q_m & \text{for } n_b<i\leq n_b+n_t; \; n_b+n_t+n_c<j \\ 0 & \text{otherwise} \end{cases}$$

where the coefficients of geometric nonlinearity — R_{mnij} resulting from modal analysis applied to $(EI_\zeta\text{-}EI_\eta)(\bar{w}''^2\theta)$ in equation (3-60), and H_{mij} resulting from modal analysis applied to $(EI_\zeta\text{-}EI_\eta)(\bar{w}''v'')$ in equation (3-60) or $(EI_\zeta\text{-}EI_\eta)(\bar{w}''\theta)''$ in equation (3-59) — are,

$$(3\text{-}157) \quad R_{mnij} = \frac{1}{c}(EI_\zeta\text{-}EI_\eta)\int \phi_{m.xx}\phi_{n.xx}\phi_i\phi_j dx$$

$$\text{for } m,n\leq n_b; \; n_b<i,j\leq n_b+n_t$$

$$(3\text{-}158) \qquad H_{mij} = (EI_\zeta - EI_\eta) \int \phi_{m,xx} \phi_i \phi_{j,xx} dx$$

$$\text{for } m \leq n_b; \ n_b < i \leq n_b + n_t; \ n_b + n_t + n_c < j$$

The modal forces are then finally obtained by integrating the aerodynamic force coefficients, as determined in Section 3.2, with the mode shapes over the span. This integral also incorporates the spanwise correction to strip theory described by Landsberger [Ref. 92] and described in Appendix C.

$$(3\text{-}159) \qquad Q_i = \tfrac{1}{2}\rho U^2 \int_0^\ell \left\{ c[C_L(x)\cos\theta_R + C_D(x)\sin\theta_R]\psi_i^w(+\tfrac{c}{4}) \right.$$

$$\left. + c^2 C_M(x)\psi_{i,y}^w(+\tfrac{c}{4}) \right\} \phi_i^w(x) dx \quad \text{for } i,j \leq n_b + n_t + n_c$$

$$(3\text{-}160) \qquad Q_i = \tfrac{1}{2}\rho U^2 \int_0^\ell c[-C_L(x)\sin\theta_R + C_D(x)\cos\theta_R]\psi_i^v(0)\phi_i^v(x) dx$$

$$\text{for } n_b + n_t + n_c < i,j$$

3.4 Pre-Flutter Analyses

3.4.1 Static Deflection Problem

The static deflection problem is formulated as an analytical model of the experimental deflection tests described in Chapter 4. For a pure force test, the cantilevered plate or wing is subjected to a concentrated load at the specimen tip $(x=\ell)$, at the elastic axis $(y=0)$. For a pure moment test, the cantilevered plate or wing is subjected to equal and opposite concentrated loads at the specimen tip $(x=\ell)$, at the leading and trailing edges $(y=\pm c/2)$. The accelerations are zero

54

for static deflection, and the real forces are point loads, so equation (3-138) for a pure force reduces to,

$$(3\text{-}161) \qquad \sum_{j=1}^{n} K_{ij}q_j = Q_i = F \phi_i(\ell)\psi_i(0) \quad i=1,\ldots,n$$

where F is the concentrated load applied at the wing tip. Similarly for a pure moment, where M is the moment applied to the wing tip, equation (3-138) reduces to,

$$(3\text{-}162) \qquad \sum_{j=1}^{n} K_{ij}q_j = Q_i = \frac{M}{c} \phi_i(\ell)[\psi_i(\tfrac{+c}{2})-\psi_i(\tfrac{-c}{2})] \quad i=1,\ldots,n$$

3.4.2 Free Vibration Problem

The free vibration problem is formulated as an analytical model of the experimental vibration tests described in Chapter 4. The problem is formulated by setting the modal forces, Q_i, equal to zero in equations (3-138) and (3-139). The equations of motion are reduced from differential form to algebraic form by assuming harmonic (sinusoidal) motion. The modal amplitudes can be expressed as,

$$(3\text{-}163) \qquad q = \tilde{q}e^{i\omega t} \quad ; \quad \ddot{q}=-\omega^2\tilde{q}e^{i\omega t}$$

where ω is the frequency. These assumptions are substituted into the differential equations of motion, (3-138), to obtain the sinusoidal equations of motion,

$$(3\text{-}164) \qquad \sum_{j=1}^{n} (-\omega^2 M_{ij} + K_{ij})\tilde{q}_j = 0 \quad i=1,\ldots,n$$

55

or, in matrix form,

$$(3-165) \quad \left[-\omega^2[M]+[K]\right]\{\bar{q}\} = \{0\}$$

Equations (3-164) and (3-165) describe an eigenvalue problem which can be solved by using a numeric eigenvalue solver, for example EISPACK™.

The linear free vibration problem about the nonlinear static deflection is carried out in the same manner as the linear free vibration problem, with the exception that the stiffness matrix is changed according to the geometrically nonlinear effects of a nonzero tip deflection. For this analysis, the tip deflection was effectuated by increasing only the static deflection in bending of the first n. '- shape, q_{1o}, in other words,

$$(3-166) \quad K_{ij} = \begin{cases} K_{ij}^{fp} + K_{ij}^{sty} + R_{11ij}q_{1o}^2 & \text{for } n_b < i,j \leq n_b + n_t \\[2ex] K_{ij}^{fp} + K_{ij}^{sty} + H_{1ij}q_{1o} & \text{for } n_b < i \leq n_b + n_t;\ n_b + n_t + n_c < j \\[2ex] K_{ij}^{fp} + K_{ij}^{sty} & \text{otherwise} \end{cases}$$

3.4.3 Two-Dimensional Aerodynamic Problem

Three approaches to the ONERA method of calculating the 2-dimensional force hysteresis were compared: a 4th-order Runge-Kutta time marching scheme, the 2-harmonic scheme described in Section 3.2.3 (denoted "non-constant" because a, r, & e are not constant through the hysteresis cycle), and a 2-harmonic scheme with constant coefficients a, r, & e (i.e. β_1, β_2, & β_3 set to zero). The pur-

poses of this comparison were, first, to determine if the harmonic schemes accurately matched the "exact" Runge-Kutta time marching scheme and, second, to determine if using non-constant coefficients provided significant improvement over constant coefficients in the harmonic schemes.

The 2-harmonic, non-constant approach is as described in Section 3.2.3. The 2-harmonic constant coefficient scheme is merely a simpler subset of the non-constant approach. Several approaches can be taken to "smearing" the nonlinear coefficients through the hysteresis cycle,

$$(3-167) \qquad a = a_0 + a_1 (\Delta C_L(<\alpha>))^2 = a_0 + a_1 (\Delta C_L(\alpha_0))^2$$

$$(3-168) \qquad a = a_0 + a_1 (<\Delta C_L(\alpha)>)^2 = a_0 + a_1 \Delta_0^2$$

$$(3-169) \qquad a = <a_0 + a_1 (\Delta C_L(\alpha))^2> = a_0 + a_1 (\Delta_0^2 + \tfrac{1}{2}\Delta_1^2 + ...)$$

Equation (3-167) bases the constant coefficient on the mean angle of attack. This approximation seems poor because there are no effects when the mean angle is below the stall angle but the amplitude of oscillation is large enough to cross into the stall region. Equation (3-168), based on the mean force deficit, and equation (3-169), based on the mean coefficient through the cycle, account for this effect, and are likely more accurate approximations. Of the two, that described by equation (3-169) is likely a better approximation, but that described by equation (3-168) was used because it was easier to implement in the current formulation: it could be achieved by simply setting $\beta_1 = \beta_2 = \beta_3 = 0$ in equations (3-123) to (3-125). It

57

should be noted that no matter which approach is taken, there is no dependence on the reduced frequency k because the values of the harmonic components of the lift deficit, Δ_0, Δ_1, & Δ_2, are independent of the reduced frequency.

The 4th-order Runge-Kutta scheme is typical for that used for second order differential equations. The stalled ONERA equation (3-64) is set up in terms of the state vector,

$$(3\text{-}170) \quad \frac{\partial}{\partial \tau} \vec{y} = \vec{f}[\vec{y}, \vec{x}] = \begin{bmatrix} 0 & 1 \\ -r & -a \end{bmatrix} \vec{y} + \begin{bmatrix} 0 & 0 \\ -r & -e \end{bmatrix} \vec{x}$$

with, $\quad \vec{y} = \begin{bmatrix} C_{z2} \\ \dot{C}_{z2} \end{bmatrix} \quad$ and $\quad \vec{x} = \begin{bmatrix} \Delta C_z \\ (\Delta \dot{C}_z) \end{bmatrix}$

and the time marching substeps are given by,

$$(3\text{-}171) \quad \begin{cases} \vec{k}_1 = \frac{\Delta(k\tau)}{k} \, \vec{f}[\vec{y}_n, \vec{x}(\tau_n)] \\[2mm] \vec{k}_2 = \frac{\Delta(k\tau)}{k} \, \vec{f}[\vec{y}_n + \frac{1}{2}\vec{k}_1, \vec{x}(\tau_n + \frac{1}{2}\frac{\Delta(k\tau)}{k})] \\[2mm] \vec{k}_3 = \frac{\Delta(k\tau)}{k} \, \vec{f}[\vec{y}_n + \frac{1}{2}\vec{k}_2, \vec{x}(\tau_n + \frac{1}{2}\frac{\Delta(k\tau)}{k})] \\[2mm] \vec{k}_4 = \frac{\Delta(k\tau)}{k} \, \vec{f}[\vec{y}_n + \vec{k}_3, \vec{x}(\tau_n + \frac{\Delta(k\tau)}{k})] \\[2mm] \vec{y}_{n+1} = \vec{y}_n + \frac{1}{6}(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4) \end{cases}$$

For the Runge-Kutta scheme, each cycle of the hysteresis was divided into 360 time steps, i.e. 1° change in $k\tau$ for each time step. In general it was found that 3 cycles were required for convergence

to a steady hysteresis loop, although more were required as the reduced frequency pushed past 0.2. Therefore, approximately 1000 time steps were required to reach convergence. Each time step required 4 substeps, and each substep required on the order of 10 multiplication operations (approx. 5 to calculate the updated a, r, e, & $\vec{x}$ values, 5 to update the state vector derivative). So, in total, the Runge-Kutta scheme takes on the order of 40,000 operations. However, executing a time step per 1° change in $k\tau$ is likely too con- servative an time marching scheme, despite the high nonlinearity of the formulation. On the order of 4,000 operations would be more realistic.

For the 2-harmonic non-constant scheme, with a single break point lift model as described in Appendix E, setting up the binomial and integral coefficients b_{ij} and I_{ij} requires 9 operations while eval- uating the harmonic coefficients of the force deficit, Δ_0, Δ_1, etc., requires another 13 operations, bringing the total number of opera- tions to approximately 20. For an n-harmonic scheme, the number of operations in evaluating the harmonic coefficients of the force deficit, Δ_0, Δ_1, etc., goes like $8(n+1)$. In evaluating the aerodynamic matrix, the worst case is like the $A_{11}^{(r,a)}$ component, which requires 30 mul- tiplication operations. There are 25 such components, thus requiring 750 operations to evaluate the entire matrix. The real number of operations is likely closer to 500 because of smaller expressions for other elements such as $A_{13}^{(r,a)}$. For an n-harmonic scheme, the num- ber of operations in evaluating the aerodynamic matrix goes like $\frac{1}{4}(n+1)^4(2n+1)^2$. The factor of $(2n+1)^2$ comes from the number of components in the matrix, the factor of $(n+1)^4$ comes from multiply-

ing out the Δ_0, Δ_1, etc. terms in ΔC_L^4 in r, while the factor of $\frac{1}{4}$ is because not all the elements of ΔC_L^4 contribute to all the components of the aerodynamic matrix. Finally, Gauss elimination to solve the matrix equation requires approximately $\frac{1}{2}(2n+1)^3$ operations, i.e. approximately 60 operations for a 2-harmonic analysis.

Therefore, in total, the 2-harmonic analysis with non-constant coefficients requires somewhere on the order of 600 operations. This is about an order of magnitude fewer operations than the Runge-Kutta scheme. However, for a 3-harmonic analysis, the $(n+1)^4(2n+1)^2$ term grows very quickly and on the order of approximately 3,000 operations are required, making the tradeoff with the Runge-Kutta scheme much less beneficial.

3.5 Flutter Analyses

3.5.1 U-g Method

As a starting point from which to investigate the full, nonlinear flutter problem, it is useful to look at the linear, small-amplitude, zero root-angle-of-attack flutter and divergence problem, which can typically be solved using what is called the U-g method.

First, because the problem is linear, the steady problem is completely uncoupled from the unsteady problem, and the two can be considered separately. So, for the unsteady problem, sinusoidal motion is first assumed,

$$(3-172) \qquad q_i = \tilde{q}_i e^{i\omega t}$$

After some algebraic manipulation, it is derived that the aerodynamic modal forces are given by,

60

$$(3\text{-}173) \quad Q_i = \omega^2 s_L \rho b^3 \left\{ \frac{[L_1 + iL_2]\ell I_{ij}}{b} \sum_{n_b} \tilde{q}_j + \frac{[L_3 + iL_4]\ell I_{ij}}{c} \sum_{n_t} \tilde{q}_j \right\} e^{i\omega t}$$

$$\text{for } i \leq n_b$$

$$(3\text{-}174) \quad Q_i = \omega^2 s_L \rho b^4 \left\{ \frac{[M_1 + iM_2]\ell I_{ij}}{bc} \sum_{n_b} \tilde{q}_j + \frac{[M_3 + iM_4]\ell I_{ij}}{c^2} \sum_{n_t} \tilde{q}_j \right\} e^{i\omega t}$$

$$\text{for } n_b < i \leq n_b + n_t$$

where the complex lift and moment terms are,

$$(3\text{-}175) \quad [L_1 + iL_2] = 1 - \frac{2i}{k} \frac{a_{oL}}{2s_L} C(k)$$

$$(3\text{-}176) \quad [L_3 + iL_4] = \frac{1}{2} - \frac{k_v L}{s_L} + \frac{2C(k)}{k^2} \frac{a_{oL}}{2s_L} + \frac{i}{k} \left\{ 1 + \left[\frac{\sigma_L}{s_L} - \frac{a_{oL}}{2s_L} \right] C(k) \right\}$$

$$(3\text{-}177) \quad [M_1 + iM_2] = \frac{1}{2} + \frac{2s_M}{s_L} - \frac{i}{k} \frac{a_{oL}}{2s_L} C(k)$$

$$(3\text{-}178) \quad [M_3 + iM_4] = \frac{1}{4} - \frac{k_v L}{2s_L} + \frac{s_M}{s_L} - \frac{2k_v M}{s_M} + \frac{C(k)}{k^2} \frac{a_{oL}}{2s_L}$$

$$+ \frac{i}{k} \left\{ \frac{1}{2} + \frac{2s_M}{s_L} + \frac{2\sigma_M}{s_L} + \frac{1}{2} \left[\frac{\sigma_L}{s_L} - \frac{a_{oL}}{2s_L} \right] C(k) \right\}$$

and where the approximation to the Theodorsen function and the mode shape integrals corrected for spanwise effects are given by,

$$(3\text{-}179) \quad C(k) = \frac{\lambda_L + \alpha_L ik}{\lambda_L + ik}$$

61

$$(3-180) \qquad I_{ij} = \cfrac{1}{1 + \cfrac{a_{oL}}{\pi AR}} \frac{1}{\ell} \int_0^{\ell} 1.11 \left[1 - \left(\frac{x}{\ell}\right)^9 \right] \phi_i \phi_j \, dx$$

Note at this point that if the linearly derived coefficients are inserted into equations (3-175) to (3-178) $[a_{oL}=2\pi; \ s_L=\pi; \ k_{vL}=\pi/2; \ \sigma_L=2\pi; \ s_M=-\pi/4; \ k_{vM}=-3\pi/16; \ \sigma_M=-\pi/4]$, then the typical 2-dimensional, linear relations, as shown in Refs. 90 and 92, are recovered.

Inserting these into the equations of motion and canceling the $e^{i\omega t}$, yields the following form of the equations of motion, written in contracted matrix form,

$$(3-181) \qquad [[K] - \omega^2[A]]\{\bar{q}\} = 0$$

where the combined aerodynamic/mass matrix is,

$$(3-182) \qquad A_{ij} = M_{ij} + s_L \rho \ell b^2 I_{ij}[L_1 + iL_2] \quad \text{for } i,j \leq n_b$$

$$(3-183) \qquad A_{ij} = M_{ij} + s_L \rho \frac{\ell b^3}{c} I_{ij}[L_3 + iL_4] \quad \text{for } i \leq n_b; \ n_b < j \leq n_b + n_t$$

$$(3-184) \qquad A_{ij} = M_{ij} + s_L \rho \frac{\ell b^3}{c} I_{ij}[M_1 + iM_2] \quad \text{for } n_b < i \leq n_b + n_t; \ j \leq n_b$$

$$(3-185) \qquad A_{ij} = M_{ij} + s_L \rho \frac{\ell b^4}{c^2} I_{ij}[M_3 + iM_4] \quad \text{for } n_b < i,j \leq n_b + n_t$$

Structural damping is then introduced into equation (3-181) by multiplying the [K] matrix by $(1+ig)$. Introducing the complex eigenvalue Z, equation (3-181) then becomes,

62

(3-186) $[[A] - [K]Z]\{\bar{q}\} = 0$

where,

(3-187) $Z \equiv \dfrac{1 + ig}{\omega^2}$

The solution method is to pick a value of reduced frequency, and solve equation (3-186) for all the corresponding complex eigenvalues Z_i (in this case by using a complex eigenvalue solver in EISPACK™). Then, for each Z, the associated frequency, structural damping, and velocity are given by,

(3-188) $\omega = \dfrac{1}{\sqrt{Re\{Z\}}}$; $g = \dfrac{Im\{Z\}}{Re\{Z\}}$; $U = \dfrac{\omega b}{k}$

The procedure is repeated for several values of the reduced frequency k, until enough values have been generated to plot a smooth U-g diagram. The divergence points are those locations where the structural damping and frequency simultaneously go to zero. The flutter points are those other locations where the structural damping goes to zero but the frequency is non-zero.

A similar analysis, involving linear aerodynamics but nonlinear structures, can also be implemented so as to incorporate geometric nonlinearities. The procedure is to first run the purely linear U-g analysis. Now, since the problem is coupled to the steady solution through the structural nonlinearities, the next step is to determine the steady deflection at the flutter velocity and desired root angle of attack. The stiffness matrix is then updated according to these steady deflections and the U-g analysis run anew. Again, the steady

63

deflections at the newly calculated flutter velocity are incorporated to update the nonlinear stiffness matrix, and thus the procedure is repeated until it converges.

3.5.2 Harmonic Balance Applied to Nonlinear Flutter Analysis

All the components of the flutter problem have been stated in differential form and now it remains to reduce the problem to an algebraic form so that it is more easily solved computationally. The general form of the differential equation describing the motion of the wing is given by equation (3-138). The left hand side of equation (3-138) contains the structural information of the problem and is described by the definitions of the stiffness and mass matrices given in Section 3.3.3. The right hand side of equation (3-138) contains the aerodynamic information of the problem, in the form of the modal forces, and is described also in Section 3.3.3.

In general the aeroelastic problem is reduced from differential form to algebraic form by assuming harmonic motion in the same manner as for the free vibration problem in Section 3.4.2 or the U-g method in Section 3.5.1. This method is acceptable for the linear flutter problem where the steady part of the solution is uncoupled from the unsteady part of the solution. However, for the nonlinear flutter problem, these two are no longer uncoupled and both must be considered at once.

First, the modal amplitudes are put into harmonic form,

$$(3-189) \qquad q_i(\tau) = q_{io} + q_{is}\sin(k\tau) + q_{ic}\cos(k\tau)$$

64

From the modal amplitudes, the angle of attack and 1/4-chord deflection at each spanwise location are also put into harmonic form,

$$(3\text{-}190) \qquad \overline{h}(x,\tau) = \overline{h}_0(x) + \overline{h}_s(x)\sin(k\tau) + \overline{h}_c(x)\cos(k\tau)$$

$$(3\text{-}191) \qquad \theta(x,\tau) = \theta_0(x) + \theta_s(x)\sin(k\tau) + \theta_c(x)\cos(k\tau)$$

where the harmonic components of the 1/4-chord deflection are,

$$(3\text{-}192) \qquad \overline{h}_0 = \sum_{i=1}^{n} \frac{q_{io}}{b}\, \phi_i^w(x)\psi_i^w(+c/4)$$

$$(3\text{-}193) \qquad \overline{h}_s = \sum_{i=1}^{n} \frac{q_{is}}{b}\, \phi_i^w(x)\psi_i^w(+c/4)$$

$$(3\text{-}194) \qquad \overline{h}_c = \sum_{i=1}^{n} \frac{q_{ic}}{b}\, \phi_i^w(x)\psi_i^w(+c/4)$$

and the harmonic components of the angle of attack are,

$$(3\text{-}195) \qquad \theta_0 = \theta_R + \sum_{i=1}^{n} q_{io}\, \phi_i^w(x)\psi_{i,y}^w(+c/4)$$

$$(3\text{-}196) \qquad \theta_s = \sum_{i=1}^{n} q_{is}\, \phi_i^w(x)\psi_{i,y}^w(+c/4)$$

$$(3\text{-}197) \qquad \theta_c = \sum_{i=1}^{n} q_{ic}\, \phi_i^w(x)\psi_{i,y}^w(+c/4)$$

Substituting equations (3-190) and (3-191) into the formula for the linear aerodynamics, equation (3-63), gives,

$$(3-198) \qquad C_{z\gamma}(x,\tau) = C_{z\gamma o}(x) + C_{z\gamma s}(x)\sin(k\tau) + C_{z\gamma c}(x)\cos(k\tau)$$

where the harmonics of the circulatory force coefficient are,

$$(3-199) \qquad C_{z\gamma o}(x) = a_{oz}\theta_o(x)$$

$$(3-200) \qquad C_{z\gamma s}(x) = F(k)L_s(x) - G(k)L_c(x)$$

$$(3-201) \qquad C_{z\gamma c}(x) = G(k)L_s(x) + F(k)L_c(x)$$

and where, in the present analysis, the F and G functions are the resulting single lag approximations to the Theodorsen function, $C(k) = F(k)+iG(k)$, namely,

$$(3-202) \qquad F(k) = \frac{\lambda_z^2 + \alpha_z k^2}{\lambda_z^2 + k^2}$$

$$(3-203) \qquad G(k) = \frac{\lambda_z k(1-\alpha_z)}{\lambda_z^2 + k^2}$$

and where the other intermediate variables are,

$$(3-204) \qquad L_s(x) = a_{oz}[\theta_s(x) + k\bar{h}_c(x)] - \sigma_z k\theta_c(x)$$

$$(3-205) \qquad L_c(x) = a_{oz}[\theta_c(x) - k\bar{h}_s(x)] + \sigma_z k\theta_s(x)$$

66

Finally, the apparent mass terms are added to give the usual harmonic form of the linear aerodynamics derived from equation (3-62),

(3-206) $\quad C_{z1o}(x) = C_{z\gamma o}(x)$

(3-207) $\quad C_{z1s}(x) = C_{z\gamma s}(x) - s_z[k\theta_c(x) - k^2 \bar{h}_s(x)] - k_{vz}k^2\theta_s(x)$

(3-208) $\quad C_{z1c}(x) = C_{z\gamma s}(x) + s_z[k\theta_s(x) + k^2 \bar{h}_c(x)] - k_{vz}k^2\theta_c(x)$

$C_{z2o}(x)$, $C_{z2s}(x)$, and $C_{z2c}(x)$ are the spanwise location values of C_{z2o}, C_{z2s1}, and C_{z2c1} as derived in Section 3.2.3. They are added to the results of the linear aerodynamics to give the combined spanwise-varying and time-varying force coefficient,

(3-209) $\quad C_z(x,\tau) = C_{zo}(x) + C_{zs}(x)\sin(k\tau) + C_{zc}(x)\cos(k\tau)$

where the harmonic components of the force coefficient are,

(3-210) $\quad C_{zo}(x) = C_{z1o}(x) + C_{z2o}(x)$

(3-211) $\quad C_{zs}(x) = C_{z1s}(x) + C_{z2s1}(x)$

(3-212) $\quad C_{zc}(x) = C_{z1c}(x) + C_{z2c1}(x)$

The harmonic form of the aerodynamic forces is then placed into equation (3-159) to give the harmonic form of the modal forces,

(3-213) $\quad Q_i(\tau) = Q_{io} + Q_{is}\sin(k\tau) + Q_{ic}\cos(k\tau)$

where the harmonic components of the modal force are,

$$(3\text{-}214) \qquad Q_{io} = \tfrac{1}{2}\rho U^2 \int_0^{\ell} \left\{ c[C_{L0}(x)\cos\theta_R + C_{D0}(x)\sin\theta_R]\psi_i(+\tfrac{c}{4}) + c^2 C_{M0}(x)\psi_{i,y}(+\tfrac{c}{4}) \right\} \phi_i(x)dx$$

$$(3\text{-}215) \qquad Q_{is} = \tfrac{1}{2}\rho U^2 \int_0^{\ell} \left\{ c[C_{Ls}(x)\cos\theta_R + C_{Ds}(x)\sin\theta_R]\psi_i(+\tfrac{c}{4}) + c^2 C_{Ms}(x)\psi_{i,y}(+\tfrac{c}{4}) \right\} \phi_i(x)dx$$

$$(3\text{-}216) \qquad Q_{ic} = \tfrac{1}{2}\rho U^2 \int_0^{\ell} \left\{ c[C_{Lc}(x)\cos\theta_R + C_{Dc}(x)\sin\theta_R]\psi_i(+\tfrac{c}{4}) + c^2 C_{Mc}(x)\psi_{i,y}(+\tfrac{c}{4}) \right\} \phi_i(x)dx$$

The general equations of motion, described in matrix form in equation (3-139), are converted into the final harmonic form by substituting the harmonic forms of the modal amplitudes and modal forces from equations (3-189) and (3-214) to (3-216),

$$(3\text{-}217) \qquad \begin{bmatrix} [K] & 0 & 0 \\ 0 & -\omega^2[M]+[K] & 0 \\ 0 & 0 & -\omega^2[M]+[K] \end{bmatrix} \begin{bmatrix} \{q_o\} \\ \{q_s\} \\ \{q_c\} \end{bmatrix} = \begin{bmatrix} \{Q_o\} \\ \{Q_s\} \\ \{Q_c\} \end{bmatrix}$$

Equation (3-217) might look deceivingly linear, but this is not so. First, because of the static aerodynamics, there is a nonlinear dependence of the mean modal forces $\{Q_o\}$ on the mean modal amplitudes $\{q_o\}$, and similarly in a quasi-steady sense for the harmonics $\{q_s\}$ & $\{q_c\}$ on $\{Q_s\}$ & $\{Q_c\}$. Second, because of the nonlinear formulation of the ONERA aerodynamic model, there is also a nonlinear

68

dependence across the harmonics — that is, there is a dependence of $\{Q_o\}$ on $\{q_s\}$ & $\{q_c\}$ and of $\{Q_s\}$ & $\{Q_c\}$ on $\{q_o\}$. Third, because of the geometric nonlinearities, there is a nonlinear dependence of the stiffness matrix $[K]$ on the mean modal amplitudes $\{q_o\}$.

3.5.3 Parameters of Analysis and Implementation

Once the governing equations are set up for the various problems of static deflection, free vibration, 2-dimensional aerodynamics, and flutter, it still remains to be determined the desired parameters to solve those problems.

As discussed in Section 3.3.2, beam bending modes and root-warped torsional modes were chosen to model the structural dynamics. Alternatively, torsional modes with warping terms ignored — i.e. pure sine mode shapes — could have been used but, while these would have accurately predicted natural frequencies, they would have overestimated static deflections because of the lack of stiffness from root warping.

In general, since the final stalled flutter problem is expected to yield a single degree of freedom motion in either the first torsional or first bending mode, it was deemed necessary to only model the first torsion and first bending modes accurately. For an uncoupled wing, such as the $[0_3/90]_S$, it would therefore only be necessary to include those two modes to get the frequencies correct. However, other factors necessitate larger numbers of modes. First, because of the bending-torsion coupling of the other layups, larger numbers of modes were required to get even the first torsion and first bending frequencies accurately. Second, because of the distributed natures of

the aerodynamic loading of the linear & nonlinear flutter analyses, larger numbers of modes were required to accurately model the distributed loads as the summation of modal forces. In other words, because of the bending-torsion coupling the mass & stiffness matrices — i.e. the left hand side of equation (3-138) — indicate a need for a larger number of modes, while the accurate modeling of the modal forces — i.e. the right hand side of equation (3-138) — also indicates a need for a larger number of modes. It was decided from these considerations that three bending and three torsion modes would suffice to accurately model the first bending and first torsion modes through all the analyses.

Again, because the flutter solution is expected to be "locked" at the torsional frequency, only one harmonic would seem necessary for the final analysis. And again, because of the coupling between harmonics inherent in the nonlinear problem, higher harmonics would seem necessary. However, the nonlinear coupling between harmonics only occurs through the aerodynamics, and hence the aerodynamic analysis by itself can use several harmonics, while the total flutter analysis can use fewer harmonics so as to save computational time.

The gross harmonic characteristics that the aerodynamic analysis is trying to capture are described graphically in Fig. 5. As already noted by Petot [Ref. 52], even within the framework of a time-marching scheme, the ONERA model does a poor job of predicting any finer details (i.e. higher harmonics) of the hysteresis cycle. In terms of harmonics, the important characteristics can be loosely

α (deg)

"in phase" – sin(kτ)   "out of phase" – cos(kτ)

α (deg)   α (deg)

"figure 8" – sin(2kτ)   "banana" – cos(2kτ)
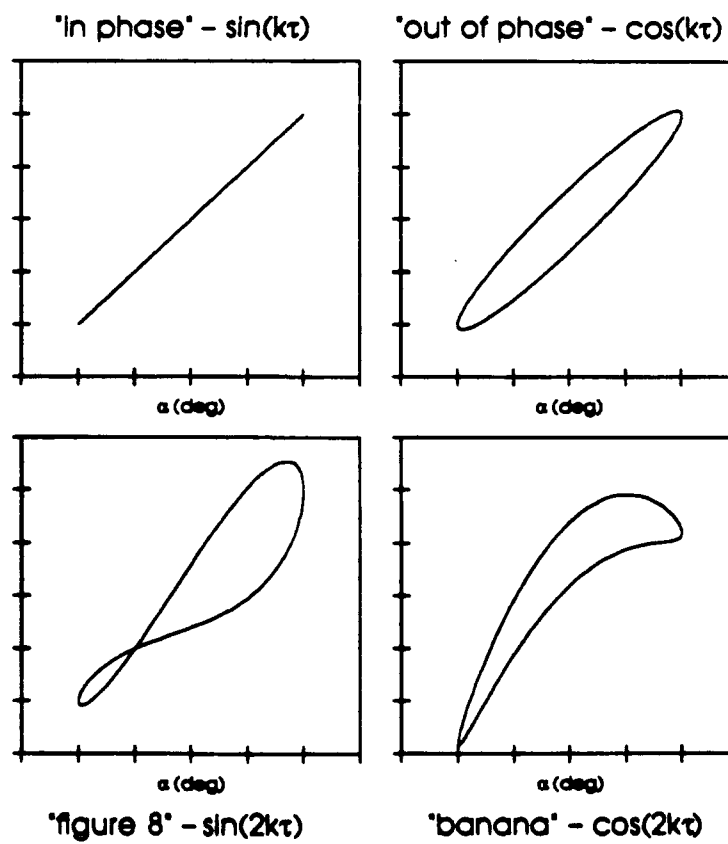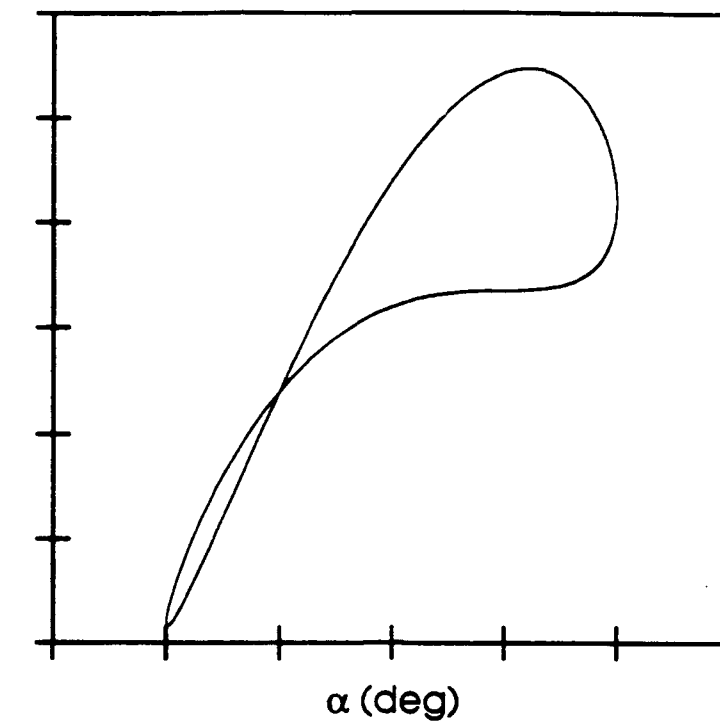
Fig. 5 Gross characteristics of force hysteresis

labeled as "in-phase" for the first sine harmonic, "out-of-phase" for the first cosine harmonic, "figure eight" for the second sine harmonic, and "banana" for the second cosine harmonic.

Physically, the first sine and second cosine harmonics are the effects of the static curve on the hysteresis — the first sine harmonic falls generally along the linear force curve, while the second cosine harmonic is affected by the deviation due to the static stalling. The first cosine and second cosine harmonics are the work terms — that is, they give the cycle its hysteretic nature and the area mapped by each cycle gives an indication of the work done by the flow on the airfoil. In general, the first harmonics can be loosely associated with the linear aerodynamics (so the first cosine term is associated with the "linear" work), while the second harmonics can be loosely associated with the nonlinear aerodynamics (so the second sine term is associated with the "nonlinear" work or work due to stalling). Obviously, these are just the broad generalities associated with each harmonic since, in reality, all the physical aspects couple into all the harmonics because of the nonlinear nature of the fluid flow. However, it seems reasonable that to capture all these physical aspects it would be necessary to incorporate at least the first two harmonics.

So, in calculating the 2-dimensional aerodynamics at each spanwise location, it was decided to use a two harmonic analysis. However, for the full flutter analysis only one harmonic was used. That is, the 2-dimensional aerodynamic analysis took a single harmonic angle of attack oscillation as its input, and calculated the resulting two harmonic components of the force coefficients (as

described in Section 3.2.3), but the flutter analysis only used the first harmonic results from these. As with the modal analysis for the structural part of the problem, it might in fact be necessary to include even higher harmonics to capture the proper physical aspects of the two lowest order harmonics. This aspect of the problem was not probed in the current investigation.

The theory described in all the previous sections was implemented using MacFortran™ code on a Macintosh IIfx at the Technology Laboratory for Advanced Composites at the Aeronautics & Astronautics Department of M.I.T. The source code of these MacFortran™ programs is listed in Appendix G. A 12-point Gauss quadrature scheme was used for all integrations that could not be easily evaluated in closed form — eg. mass & stiffness integrals (equations (3-146) to (3-158)), modal force integrals (equations (3-214) to (3-216)), or U-g method aerodynamic integrals (equation (3-181)). The Gauss points and weights are listed in the include file "GAUSS.INC" in Appendix G. It is known that in order to accurately integrate higher order polynomials, the Gauss quadrature scheme places more emphasis toward endpoints with higher clustering and weighting there — this is appropriate for the current analysis because of the evanescent contributions to the mode shapes (sinh & cosh) and because of the aerodynamic force dropoff at the tip.

73

Chapter IV
Experiment

4.1 Test Specimen Sizing

The objectives of redesigning the test specimens were twofold: first, to decrease the linear flutter velocity to within the limits of the available wind tunnel facilities (approx. 30 m/s); second, to increase the Reynolds number at flutter to a value closer to that for which the ONERA aerodynamic model was developed. These two objectives led to a single overall objective: keep the linear flutter velocity just under the wind tunnel limit — thus pushing the Reynolds number as high as possible by means of the wind velocity — and increase the chord as much as possible. These design objectives were constrained by several limitations.

Choice of layup — to ensure a wide enough range of bending-torsion coupling, and because of the manufacturing errors inherent in the layup procedure, we chose to limit ourselves to ply angles in increments of 15° only; moreover, we desired to choose our layups such that one had a divergence velocity above the wind tunnel limit, a second had a divergence velocity very near its flutter velocity, and a third had a divergence velocity below its flutter velocity, but not below the lower wind tunnel limit (approx. 10 m/s).

Number of plies — in general, an increase in number of plies increased the bending and torsional stiffnesses of the wings. Therefore, on the one hand, we wished to decrease the stiffness so as to keep the flutter velocity within the wind tunnel limits and to allow the wing to twist enough to reach the stall angle. On the other

hand, we wished to keep the stiffness high enough so that gravity bending effects would be negligible and so that bending deflections would not diverge too quickly.

Span — it was desirable to increase the span as much as possible, for purposes of observability, but we were restricted by the size of the wind tunnel (approx. 3 feet, to allow for wall effects) and by the flexibility of the specimen: we did not want it to become so long and thin that it would break at divergence or flutter.

Chord — again, it was desirable to increase the chord as much as possible, so as to increase the Reynolds number, but it was also necessary to keep the chord small enough such that the wing would be torsionally soft enough to reach flutter within the wind tunnel limits. Both the chord and the span were also limited by the size of the available autoclave at the Technology Laboratory for Advanced Composites manufacturing facility.

Most of these goals were achieved, and improved upon the layups of Refs. 94 and 95, with the following designs: layups of $[0_3/90]_S$, $[+15_2/0_2]_S$, and $[-15_2/0_2]_S$ with the span doubled from 1 ft. to approximately 2 ft., and the half-span aspect ratio kept at 4.

4.2 Test Specimen Preparation

The test specimens were constructed from Hercules AS4/3501-6 graphite/epoxy prepreg tape from Lot No. 5874-2, Spool No. 4 and Lot No. 6075-2, Spool No. 5D, using the standard TELAC manufacturing procedure [Ref. 96]. The laminates and curing materials were arranged on an aluminium curing plate as shown in

76
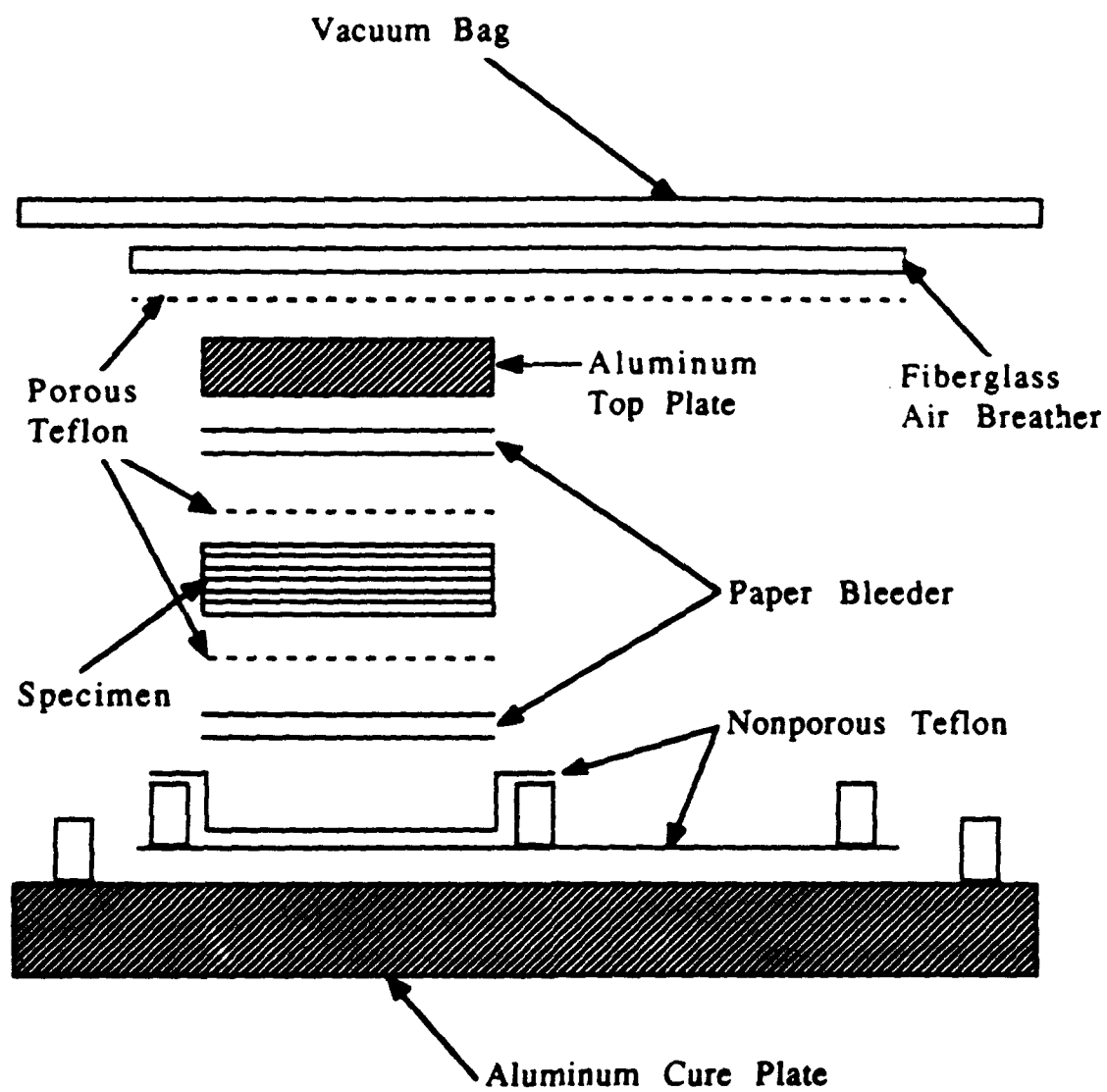
Fig. 6 Cross-section of symmetric curing assembly

Fig. 6 and cured in a Baron model BAC-35 autoclave using the standard TELAC curing cycle described by Fig. 7. After curing, the laminates were post-cured in a forced air circulation oven at 350° F for eight hours. After post-curing, rectangular test specimens 584 mm (23 in) long and 140 mm (5.5 in) wide were cut from the laminates using a diamond-coated cutting wheel mounted on an automatic feed, milling machine.

Loading tabs 152 mm (6 in) by 25.4 mm (1 in) were machined from 3.2 mm (1/8 in) aluminum plate and bonded to the base of each test specimen with FM-123-2 film adhesive, cured using the standard TELAC bond curing cycle. The loading tabs were intended to aid in aligning the test specimen in the clamping fixture and to prevent damage to the plate surface fibers.

To get an indication of the lateral deflections, strain gauges were attached to the base of each test specimen at the midchord, as shown in Fig. 8. Two Micro-Measurement EA-06-125AD-120 strain gauges, from Lot No. R-A38AD605 with a gauge factor of 2.055, were attached to both sides of each specimen near the root to measure bending strain. Two Micro-Measurement EA-06-250TK-120 strain gauges, from Lot No. R-A38AD399 with a gauge factor of 2.02, were attached to both sides of each specimen near the root to measure torsion strain. The two bending gauges were wired together as a two-arm bridge circuit with three external lead wires. The two torsion gauges were wired together as a four-arm bridge circuit with four external lead wires. Wiring the strain gauges in this manner provided automatic temperature compensation. Finally, the gauges

78

AUTOCLAVE TEMPERATURE (^{0}F)

350
240
150
RT
~70
0

10   35   95 115   235   275 280

TIME (mins)

PLUS 8 HOUR POSTCURE AT 350^{0}F

AUTOCLAVE PRESSURE (psia)

85

0

10   275 280

TIME (mins)

VACUUM (psia)

14

0

280

TIME (mins)

Fig. 7 TELAC cure cycle

STYROFOAM FAIRINGS

GRAPHITE/EPOXY
6 PLY LAMINATE

ALUMINIUM LOADING TABS

2.25"

1.5"

5.5"

2.75"

20"

23"

1"
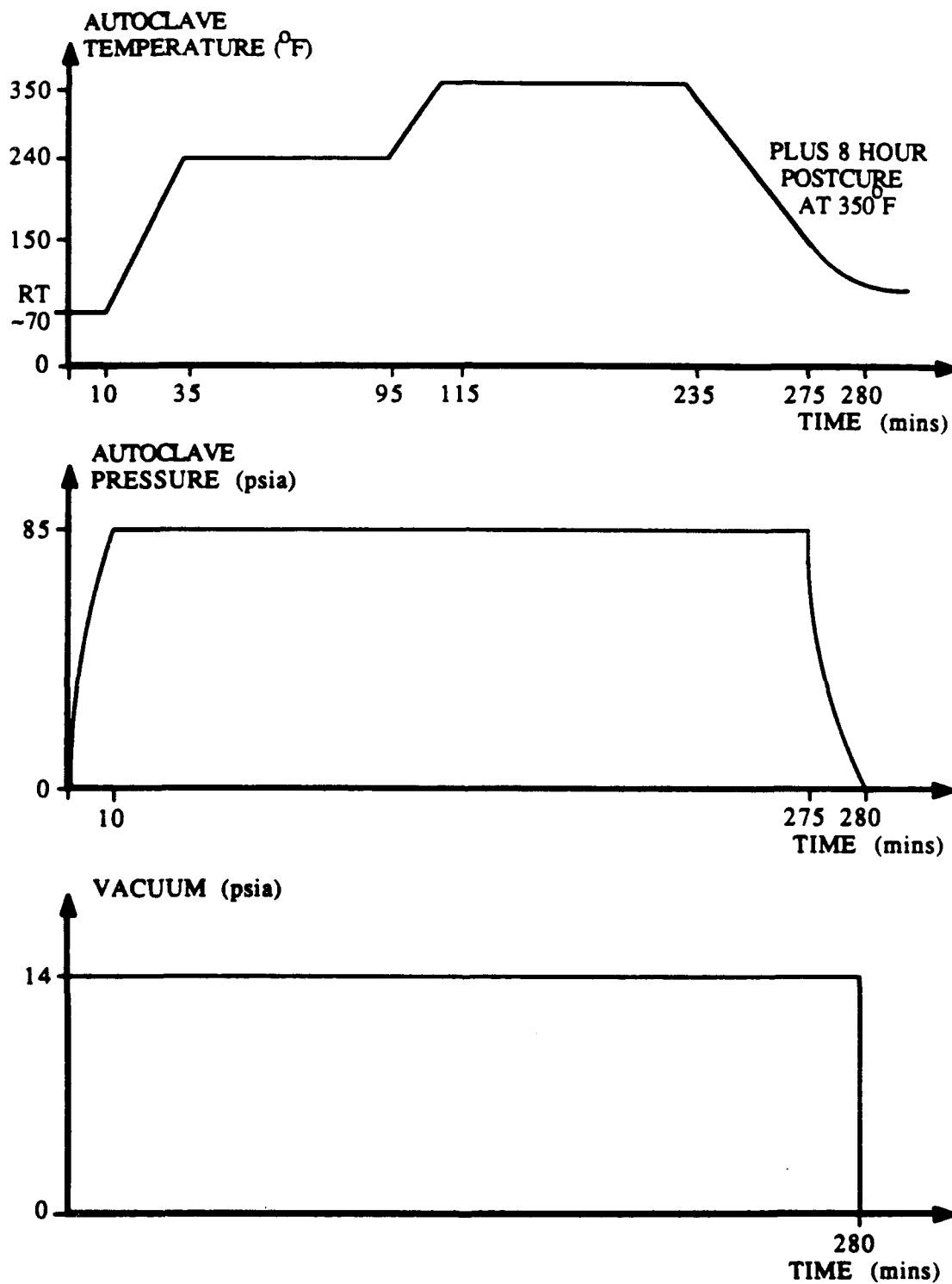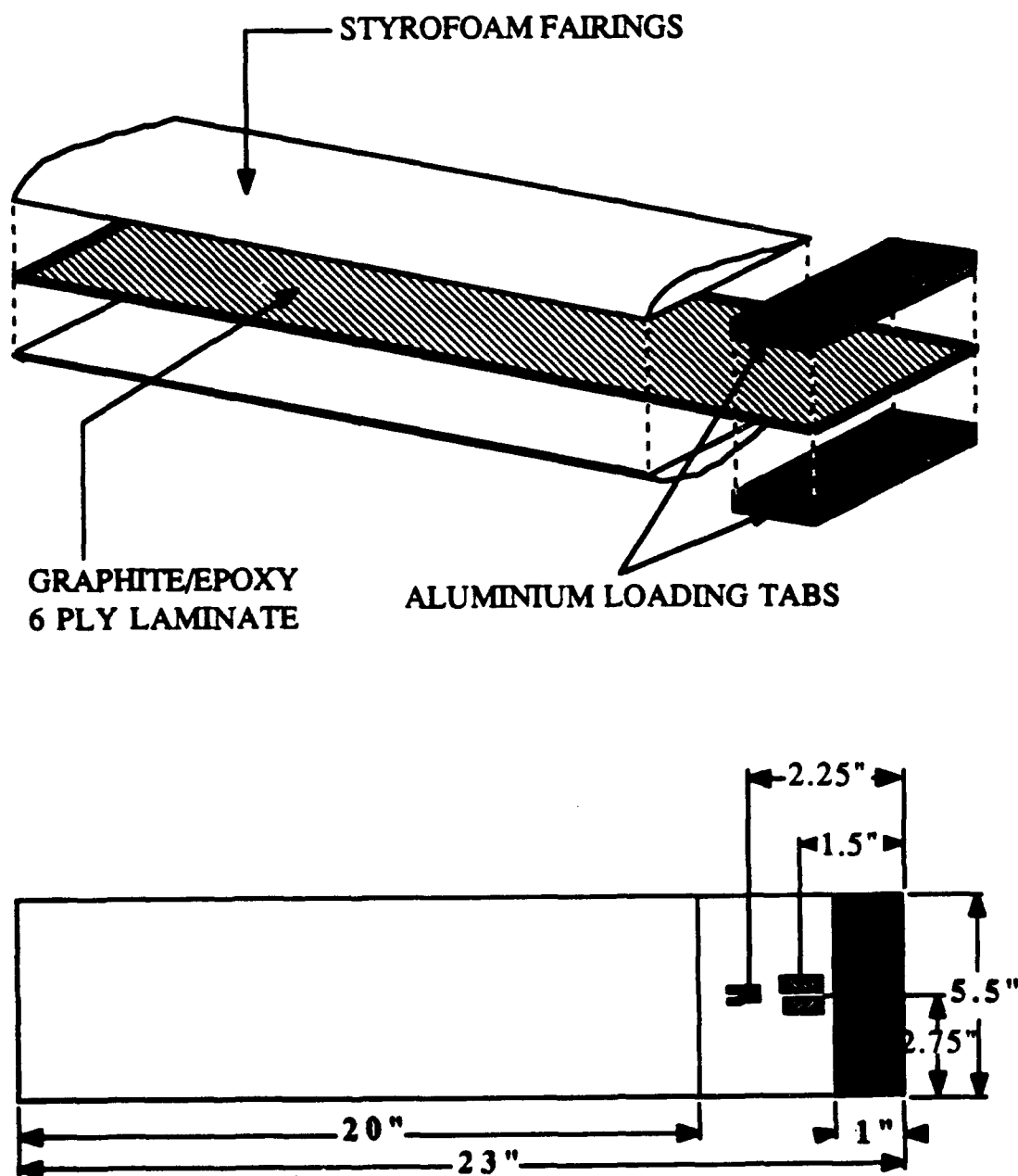
Fig. 8 Wing construction and specimen dimensions

80

and exposed wiring were coated with Micro-Measurement M-Coat A, an air-drying polyurethane.

The NACA 0012 fairings were cut from 508 mm (20 in) blocks of styrofoam using a computer controlled hot wire cutter and were then epoxied to the top and bottom of the graphite/epoxy plates.

4.3 Static Deflection Tests

The static deflection test setup (see Fig. 9) consisted of a clamping device bolted to a large aluminum table (the "optics bench" at M.I.T.'s Space Engineering Research Center). Two low friction pulleys were attached to vertical rods such that a force or moment could be applied to the test specimen at its tip. Rulers, graduated in millimeters, were attached to Dexion angle-iron to facilitate measuring the test specimens' tip deflections. Threads, routed over the pulleys and attached to weights, could be attached at any point along the wooden dowels so as to transfer either a force or a moment to the test specimen.

The deflection indicator was aligned with the tip of the test specimen and the test specimen clamped in the vise. For the tip force test, the pulleys were aligned with the plate midchord and threads from the center of the wooden dowels were routed over the pulleys. Weights in increments of 100 grams were successively attached to the threads, first to give positive deflections, then to give negative deflections. As each weight was attached, the readings from both pointers were recorded, along with the applied weight and the measured bending and torsion strains.

LOW
FRICTION
PULLEY

TIP
MEASURING
RULER

MASS

CANTILEVERED
GR/EP SPECIMEN
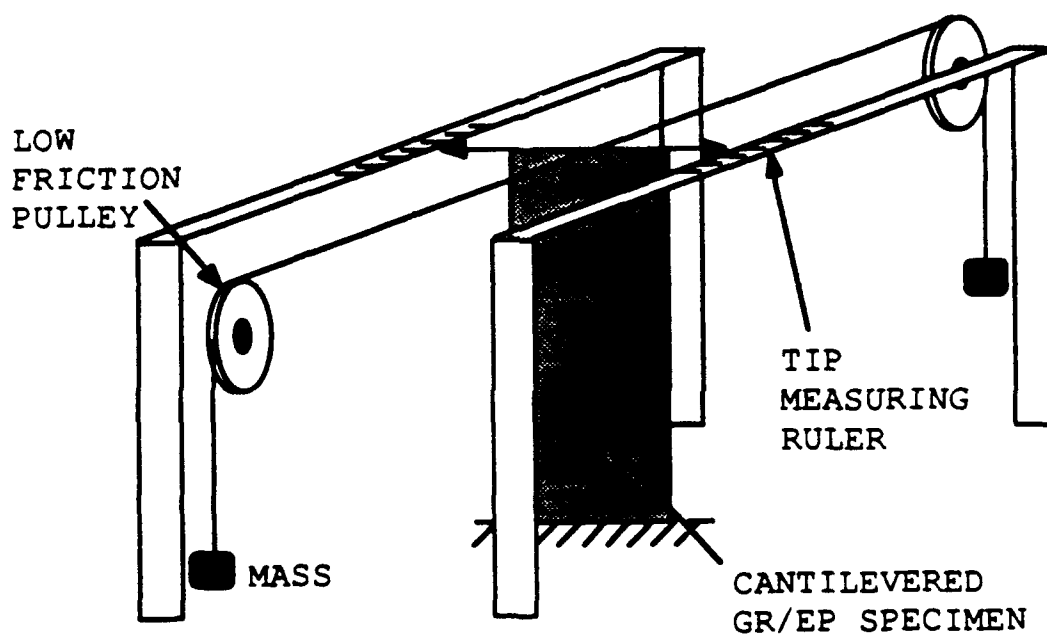
Fig. 9 Static deflection test setup

82

Next, the pulleys were aligned with the leading and trailing edge of the plate tip and the threads routed from the plate corners over the pulleys, so as to produce a positive moment when equal weights were attached. Weights of 20 gram increments were successively attached to each thread of the couple, and readings from the pointers and the strain gauges were again recorded along with the applied weights. The pulleys were then switched to diagonal opposites of the plate so that negative moments could be applied, and the same procedure applied.

For each data point, the lateral deflection of the elastic axis and the rotation about the elastic axis were calculated from the pointer measurements. The lateral and angular deflections were plotted versus applied tip force for each test specimen, and compared against the Rayleigh-Ritz analysis. Similarly, the lateral and angular deflections were plotted versus applied tip moment for each test specimen, and compared against the same analysis. Linearized fits between lateral deflection and bending strain, and between angular deflection and torsion strain, were conducted so that a linear relation could be later applied to the flutter tests. The results of the static deflection tests are discussed in Section 5.1.1.

4.4 Free Vibration Tests

"Pluck" tests were conducted before each flutter test to verify the free vibration frequencies of the wings. After the wings had been clamped vertically in the specimen stand, they would either be sharply tapped or given a brief, sharp torsional force, thus exciting several of the lower bending and torsion modes. The strain gauges

83

were wired to 2120 Strain Gauge Amplifiers, with a two-arm D.C. bridge for the bending gauges and a four-arm D.C. bridge for the torsion gauges, and their readings were recorded on floppy disk using a Nicolet digital oscilloscope.

Later, these signals were passed through a Fourier analyzer so as to decompose the frequency content of the signal. Since the free vibration modes would presumably have been excited by the sharp taps, the peaks of the resulting frequency spectrum of the FFT would correspond to the natural frequencies of the specimens. The signals were made up of 2048 data points taken at 5 ms between data points, thus corresponding to 2048 data points over 10 seconds or, approximately, a frequency range of 0-100 Hz with a frequency resolution of 0.1 Hz. The results of the free vibration tests are discussed in Section 5.1.2.

4.5 Wind Tunnel Tests

All wind tunnel tests were conducted in the M.I.T. Department of Aeronautics and Astronautics acoustic wind tunnel. The acoustic wind tunnel is a continuous flow tunnel with a 1.5 m (5 ft) x 2.3 m (7.5 ft) free jet test section 2.3 m (7.5 ft) long. The tunnel was powered by a 100 HP motor giving it a continuously variable velocity range of 0 m/s to 30 m/s (0 ft/sec to 105 ft/sec). The tunnel control panel was located inside the chamber and the velocity was controlled by two knobs (coarse and fine speed control).

The test setup, shown in Fig. 10, consisted of a turntable machined from aluminum, mounted on a 914 mm (36 in) tall, cylindrical pedestal made of 51 mm (2 in) thick steel pipe, 305 mm

(24 in) in diameter. The pedestal was mounted to the floor of the wind tunnel section. A wooden cover disk 762 mm (30 in) in diameter was used to ensure the pedestal did not affect the flow over the test specimen, and thus provided smooth airflow past the test specimen. A pointer attached to the free rotating portion of the turntable, and an angle indicator attached to the fixed base of the turntable, provided a consistent means of reading the angle of attack of the test specimen.

The bending and torsion strain gauges were wired to a terminal strip attached to the fixed pedestal, which was in turn wired to 2120 Strain Gauge Amplifiers. The amplifiers had a two-arm D.C. bridge installed in channel 1 for the bending gauges and a four-arm D.C. bridge installed in channel 2 for the torsion gauges. The bending and torsion outputs from the Strain Gauge Amplifiers were fed to a Nicolet Digital Oscilloscope where the signals could be recorded on floppy disk. Visual data was recorded by placing a mirror at a 45° angle above the test setup, and recording onto 8mm videotape the overhead view of the tip deflections. For sinusoidal flutter motion, a strobe light was used to help visualize the oscillations.

The procedure for running the flutter tests was done in two steps. First, the flutter boundary was determined. The root angle of attack was increased by increments of 1°, and at each angle of attack the velocity was slowly increased until the onset of flutter was observed, marked by a visually noticeable amplitude of oscillation and a clear frequency of oscillation (as distinguishable from wind tunnel turbulence). Next, static and flutter data was taken at 1°, 5°, 10°, and 15° root angles of attack. At each of these, the velocity was

85

increased in increments of 1 m/s and visual and strain gauge data taken at each velocity value. This was continued up to the flutter boundary. The procedure was also continued past the flutter boundary, for larger amplitudes of oscillation, but care was taken not to remain too long past the flutter boundary, for fear of damaging the specimens.
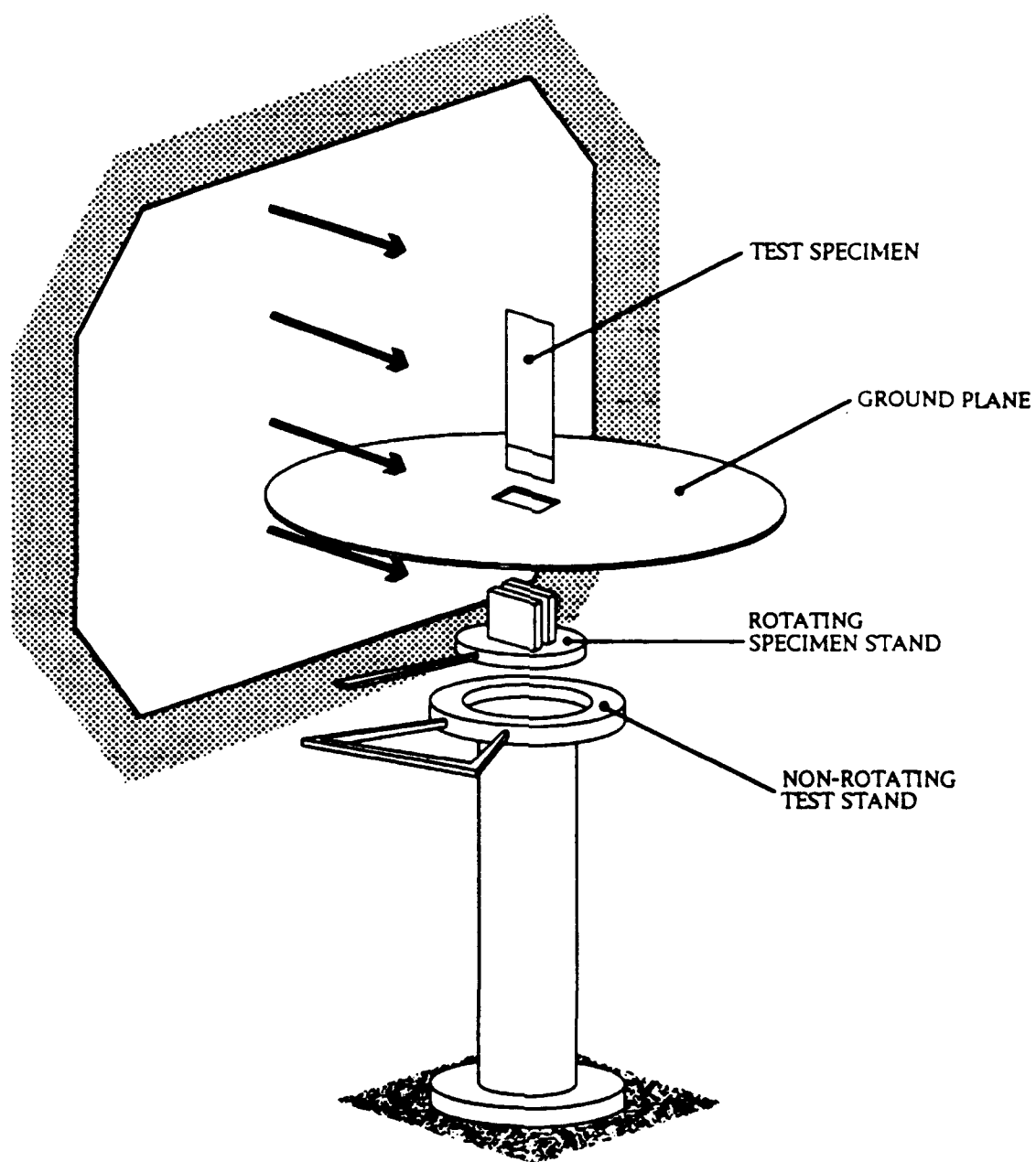
Fig. 10 Wind tunnel test stand

Labels in figure:
- TEST SPECIMEN
- GROUND PLANE
- ROTATING SPECIMEN STAND
- NON-ROTATING TEST STAND

88

Chapter V
Results and Discussion

The Results & Discussion chapter is divided into two sections: pre-flutter results and flutter analysis. The objectives of the section on pre-flutter results is to verify individually the various components that make up the flutter analysis — stiffness properties, mass properties, effects of geometric nonlinearities, and 2-dimensional aerodynamics — using the various methods of static deflection tests, *in vacuo* dynamics (i.e. free vibration) at zero deflection, and free vibration with tip deflection. Once the individual components of the flutter analysis are verified, the objective of the section on flutter analysis is to combine these components and move from linear analysis to fully nonlinear analysis in a stepwise fashion. This section starts by approaching the problem using a fully linear U-g analysis; adds nonlinearity through large mean deflections and angles of attack while keeping oscillation amplitudes small (i.e. flutter boundary analysis); then adds another level of nonlinearity by considering large amplitudes of oscillation.

5.1 Pre-Flutter Results

5.1.1 Static Deflections

The experimental results of the static deflection tests for the $[0_3/90]_S$, $[+15_2/0_2]_S$, and $[-15_2/0_2]_S$ laminates with NACA 0012 styrofoam fairings are compared in Figs. 11 to 13 with the Rayleigh-Ritz analysis described in Section 3.4.1. These figures show excellent agreement between experiment and analysis.

Because of the symmetry of the mass and stiffness matrices — i.e. the bending-torsion coupling terms are equal to the torsion-bending coupling terms — the analytic force-vs.-angle $dF/d\theta$ and moment-vs.-deflection dM/dh slopes should be equal when expressed in the same units. For example, for the $[0_3/90]_S$ layup, both these analytic slopes are approximately zero (with some slight bending-torsion coupling due to the chordwise asymmetry of the styrofoam fairings), while for the $[+15_2/0_2]_S$ layup the analytic values as seen on Fig. 12 are $dF/d\theta = -.527$ N/deg and $dM/dh = -.302$ Nm/cm or, in equivalent units, -30.2 N/rad and -30.2 Nm/m respectively. The magnitudes of the analytically derived slopes for the $[-15_2/0_2]_S$ layup are almost exactly the same as for the $[+15_2/0_2]_S$ layup, just opposite in sign, as is readily seen by comparing Figs. 12 and 13 — again, the slight difference in absolute values is due to the slight chordwise asymmetry of the styrofoam fairings. The fact that the experimental values match so closely the analytic values indicates that the assumption of symmetry is valid. In other words, neither manufacturing defects (such as misaligned layup, or variability of ply thickness) nor styrofoam asymmetry adversely affect the symmetry assumption.

The force-vs.-deflection and moment-vs.-deflection experimental slopes are almost purely linear up to the expected maximum deflection that would be encountered during a flutter experiment, i.e. 15-20 cm or about 40% of the span. However, the force-vs.-twist and moment-vs.-twist experimental slopes show hardening characteristics as compared against the linear analysis, that is, the experi-

90

mental values require a greater force or moment to produce the same twist than would be predicted by the analysis.

These hardening effects appear for the $[0_3/90]_S$ layup below -4° and above +6°, for the $[+15_2/0_2]_S$ layup below -6° and above +6°, and for the $[-15_2/0_2]_S$ layup below -6° and above +6°. For a tip twist of 9°, the discrepancy between the required moment from analysis and from experiment is consistently about 30% (relative to the experimental values) for all of the layups. This observed hardening effect, which is not accounted for in the analysis, would imply that in a flutter analysis, once the magnitude of the twist exceeded 6°, that the expected experimental phenomenon would harden more quickly that the analytic phenomenon predicted by the current analysis. Typical twist values encountered in flutter for the current investigation are on the order of 10° for light stall flutter (root angles of attack below the stall angle), and on the order of 5° for deep stall flutter (root angles of attack at or above the stall angle), and therefore it is expected that some structural nonlinearity will be unaccounted for by the analysis near linear flutter. Also, an amplitude of oscillation in flutter of 6° is generally considered "moderate" in terms of the current aerodynamic analysis, while anything larger is considered "large". That is, experimentally all amplitudes from small to large are observed (as can be seen from the figures in Section 5.2.3), but the aerodynamic analysis described in Chapter 3 is assumed to be valid for moderate amplitudes and to break down for large amplitudes.

Within the range of -6° to +6° twist, the analytic prediction of the twist from the applied forces generally falls within less than 1°

of the experiment, which is less than the experimental error for the measured angles. Likewise, the analytic prediction of the deflections from the applied forces falls within the experimental error over the entire range that would be expected in flutter. Therefore, comparison of the experimental static deflections with their analytic predictions indicates that the stiffness properties of the wings are accurately predicted for most of the range for which the flutter analysis is expected to be applied, and will only lack some cubic stiffening in torsion at flutter values near divergence or at large amplitudes of oscillation.
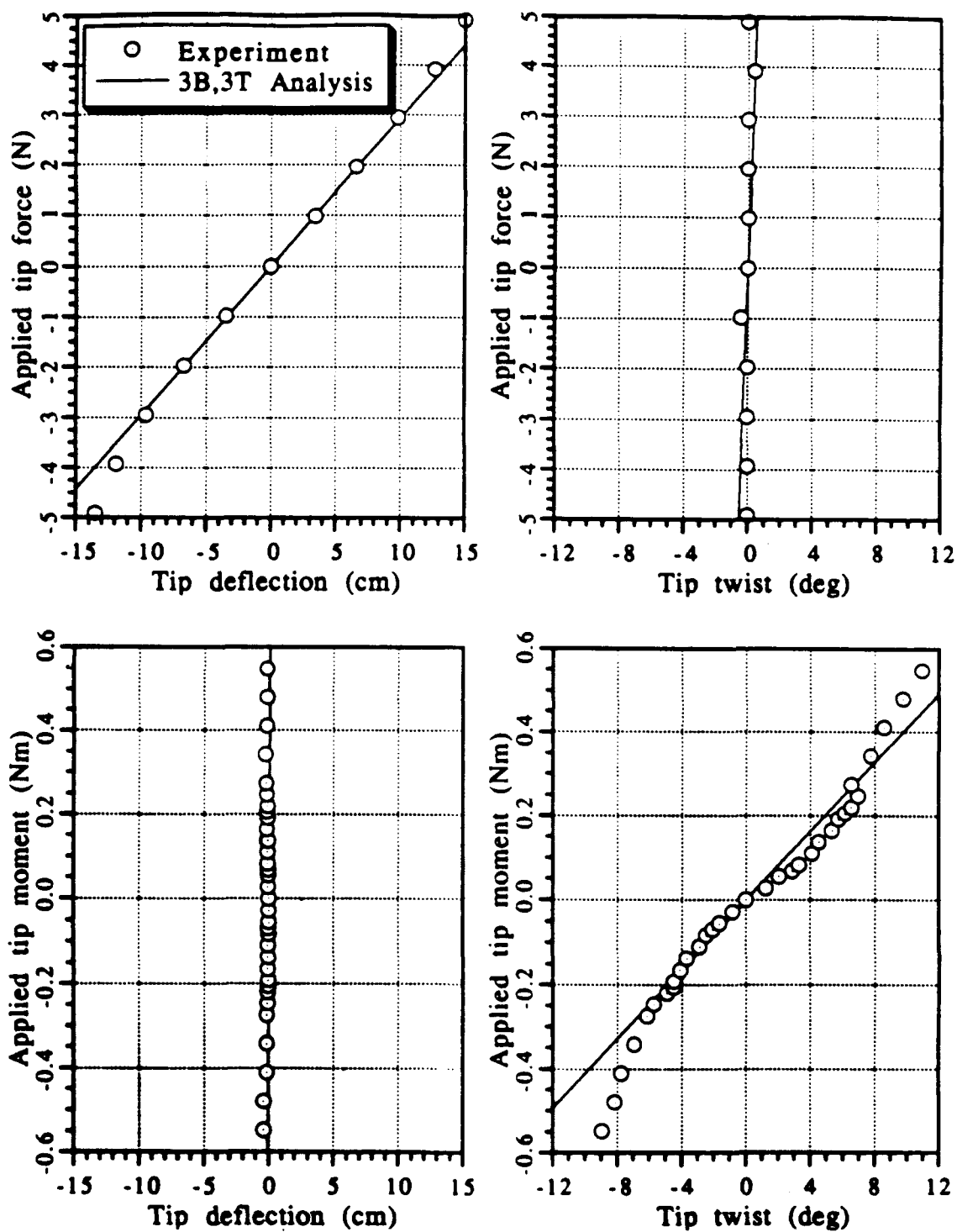
Fig. 11 $[0_3/90]_S$ wing static deflection results
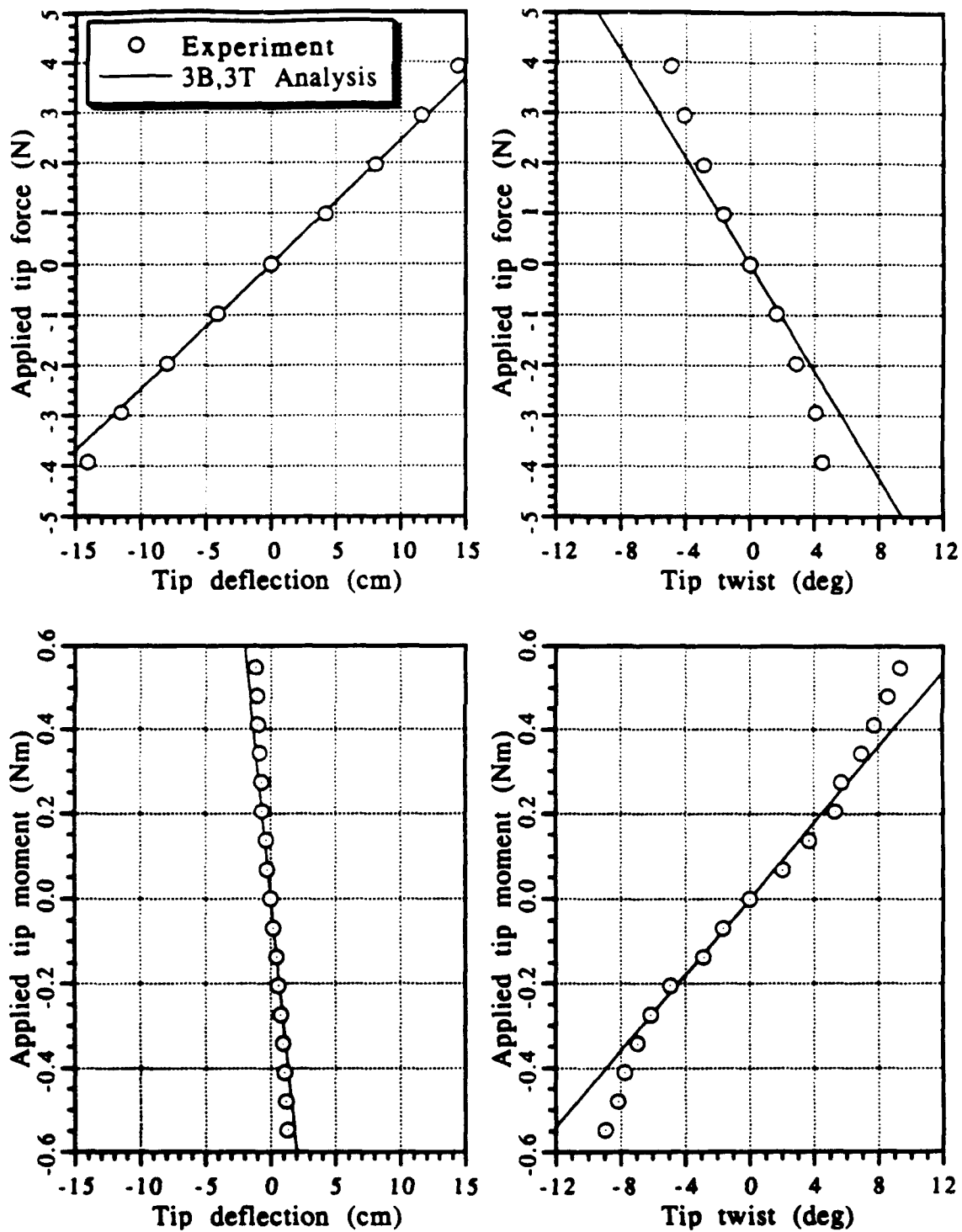
Fig. 12 $[+15_2/0_2]_S$ wing static deflection results

94
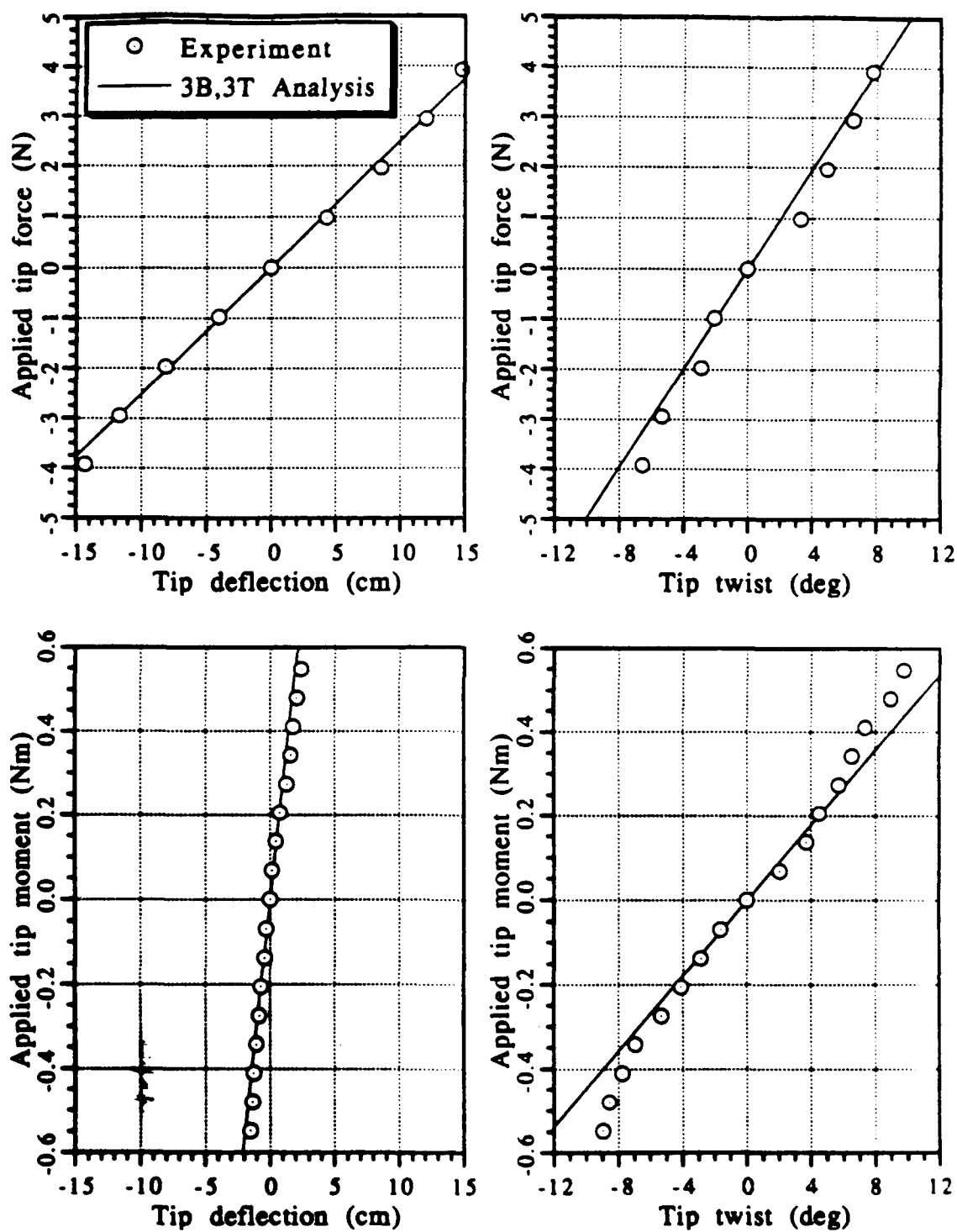
Fig. 13 $[-15_2/0_2]_S$ **wing static deflection results**

5.1.2 Linear Free Vibration

Natural vibration frequencies for the NACA 0012 wings without tip deflection were determined both experimentally and analytically, and are tabulated in Table 2. Although these are listed as 1st bending (1B), 1st torsion (1T), et cetera, with highly coupled laminates this distinction becomes much less meaningful because of the high bending-torsion coupling. Figs. 14 and 15 show the uncoupled mode shapes of the $[0_3/90]_S$ layup and the coupled mode shapes of the $[+15_2/0_2]_S$ layup.

The experimental frequencies show excellent agreement with the analysis for the first & second bending modes and the first torsional mode. While the percentage errors for the first bending mode frequencies might seem high, it should be noted that the frequency resolution of the Fast Fourier Transform that was applied to the signal from the pluck test (see Section 4.4) was about 0.1 Hz, and therefore a large part of the 0.3 Hz discrepancy might be accounted for by this experimental error. The analysis could not be used as comparison against experiment for the higher modes because it was found to be too difficult to significantly excite the higher modes by the pluck test described in Section 4.4, but they are less important in the final analysis since they are only intended as corrections to the more important lower modes. One would speculate that the higher modes are likely less well predicted because the styrofoam fairing is discontinuous (i.e. it does not cover the flat plate for a short span near the root), and root warping terms become more significant for

higher modes because of the exponentially growing evanescent terms (i.e. the cosh & sinh contributions to the mode shape).

It is noteworthy that the frequencies of all the layups fall in almost the same range, both analytically and experimentally. For example, all the first bending frequencies are clustered near 4 Hz, all the first torsion frequencies are clustered near 23 Hz, and all the second bending frequencies are clustered near 27 Hz. The choice of layups, picked so as to keep the flutter and divergence speeds within the limited range of the laminar flow of the wind tunnel, is likely the source of this coincidence.

The reasons why these frequencies remain relatively unchanged are several. First, as indicated in Appendix B, the D_{11} values are all clustered near 10 Nm for all the layups, so it would be expected that the first and second bending frequencies would remain relatively unchanged. Second, again as indicated in Appendix B, the β values for all the layups, indicating the effect of the root warping on the torsional frequency, all fall below 0.05, which leads to a change in the first torsional frequency of less than 10% [Jensen, Ref. 70]. Third, as indicated in Appendix B, while the D_{66} values vary by a factor of almost two, these values are for the flat plates only. From Appendix A one sees that while the shear modulus of the styrofoam is three orders of magnitude smaller than the shear modulus of the graphite/epoxy (8 MPa as compared to 5.3 GPa), the graphite/epoxy is only 1 mm thick while the styrofoam at its widest is 12% of the chord or 17 mm thick. Therefore, because of the z^3 stiffness dependency on the thickness, the torsional stiffness contribution from the styrofoam is comparable to that of the

97

graphite/epoxy. (This would not be true of the bending stiffness, since the longitudinal modulus of the graphite/epoxy is about *four* orders of magnitude greater than that of the styrofoam.) As suggested in Section 3.3.3, the contribution to the torsional stiffness from the styrofoam fairings may be accounted for by "smearing" the styrofoam evenly across the chord at approximately 80% of its maximum thickness — this would lead to an "equivalent" D_{66} contribution from the styrofoam fairings of approximately 1.2 Nm, which is greater than the D_{66} contributions from the graphite/epoxy for all of the layups (see Appendix B). For this reason, the first torsional frequencies remain about the same, despite the difference in the flat plate torsional properties between layups.

This small variance in frequency has significant implication on the linear U-g flutter analysis: if there are noticeable variations in flutter results among the different layups, then these cannot be attributed to frequency coalescence (since the natural frequencies are all almost the same), but must be dependent on the bending-torsion effect on mode shapes as well.

98

		NACA 0012 wings		
		Experiment	Analysis	% error
$[0_3/90]_S$	1B	4.0	4.3	7.5
	1T	21.4	24.6	15.0
	2B	27.1	27.2	0.4
$[+15_2/0_2]_S$	1B	3.6	3.9	8.3
	1T	22.7	23.5	3.5
	2B	27.1	28.6	5.5
$[-15_2/0_2]_S$	1B	3.6	4.0	11.1
	1T	24.5	24.1	-1.6
	2B	27.4	27.8	1.5

Table 2. Free vibration frequencies (all values in Hz)

99
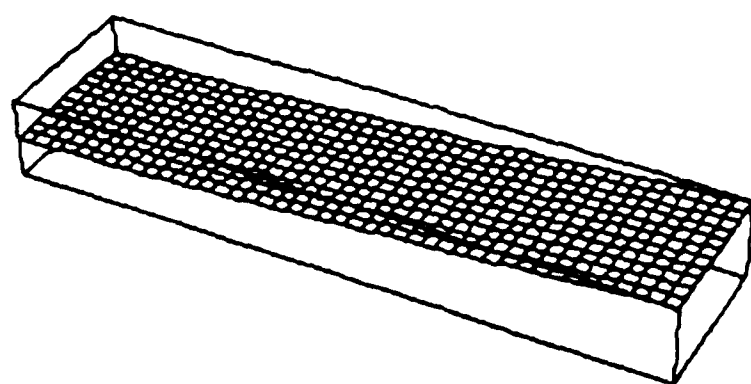
1B: 4.3 Hz
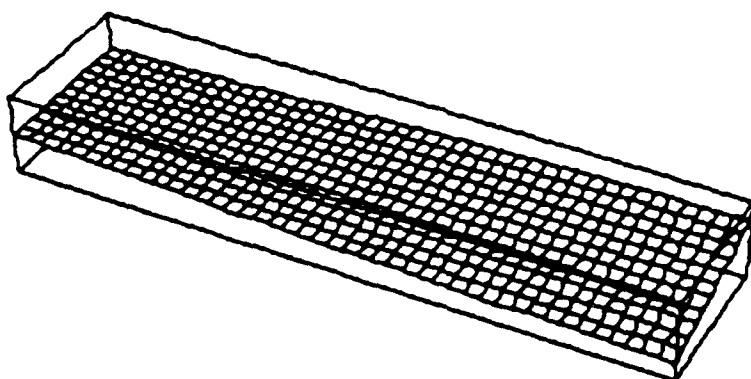
1T: 24.6 Hz
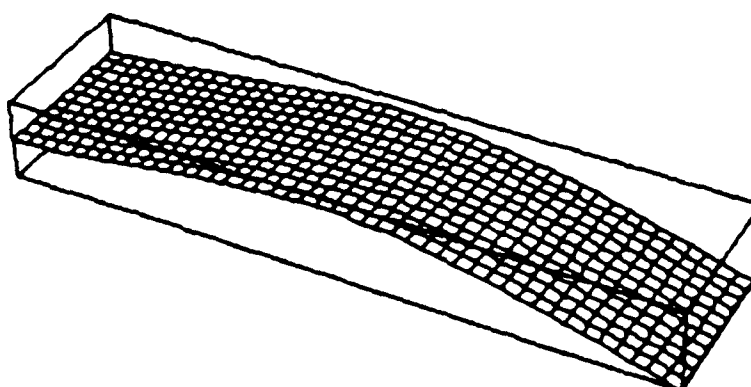
2B: 27.2 Hz
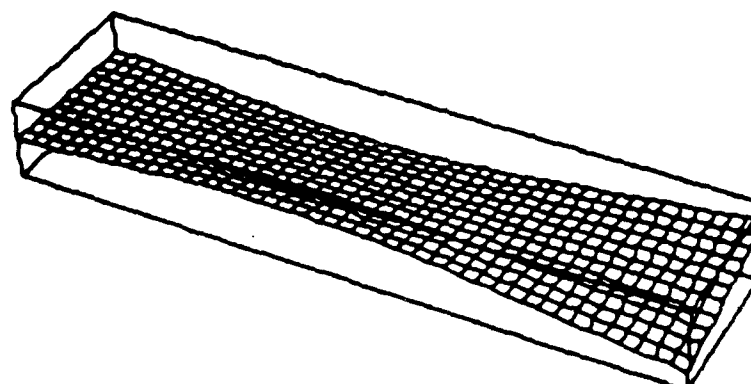
2T: 79.2 Hz

Fig. 14 $[0_3/90]_S$ analytic free vibration mode shapes

100
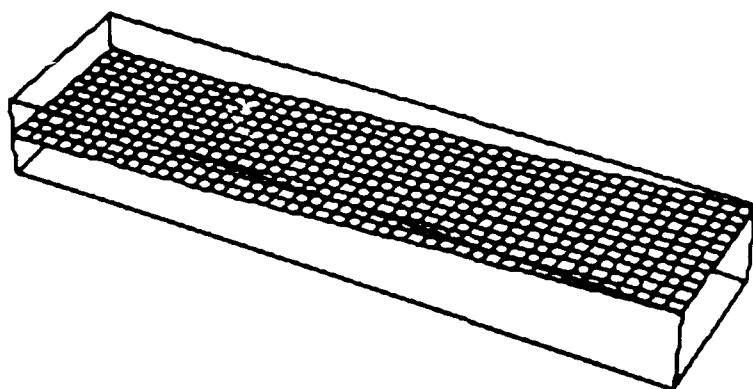
1B: 3.9 Hz
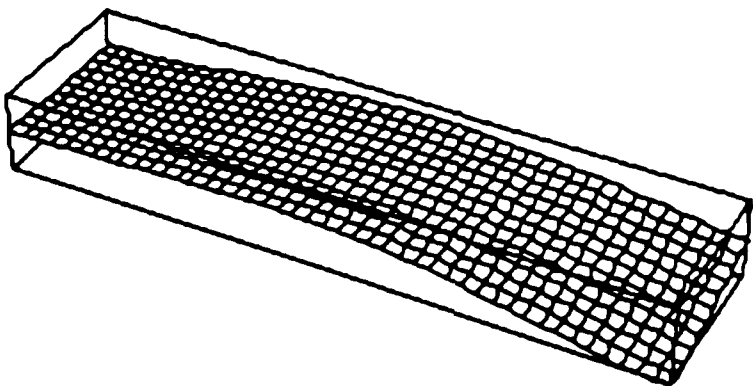
1T: 23.5 Hz
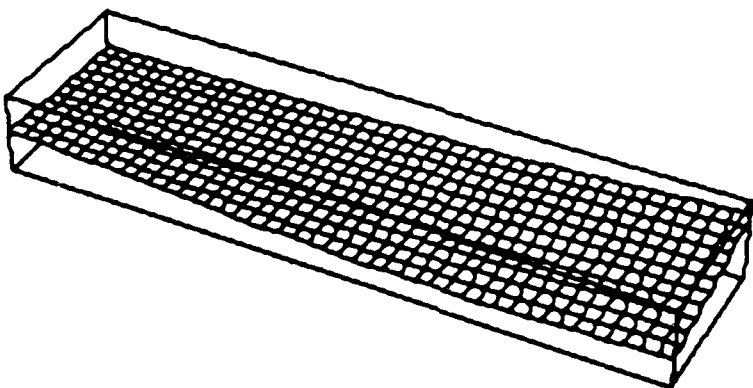
2B: 28.6 Hz
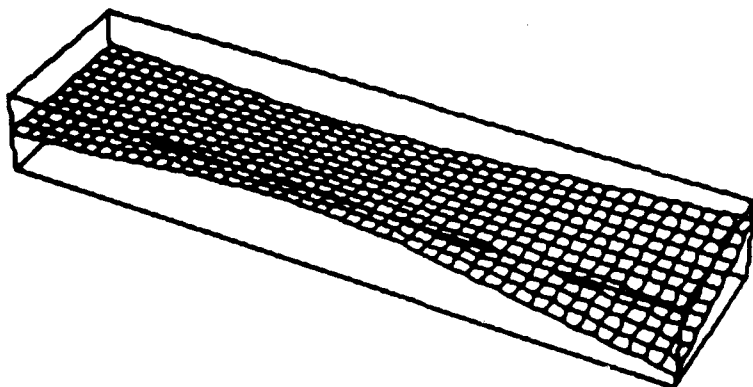
2T: 89.8 Hz

Fig. 15 $[+15_2/0_2]_S$ analytic free vibration mode shapes

5.1.3 Nonlinear Free Vibration

The nonlinear natural vibration frequencies for the NACA 0012 wings were determined analytically over a range of tip deflections from 0 cm to 20 cm, assumed to result from a distributed load. The analysis, as described in Section 3.4.2, was carried out using varying numbers of bending, torsion, and fore-&-aft modes, to determine how many modes would be required to accurately describe the frequency variation over the desired range of tip deflections. These various analyses were compared against a finite difference method that exactly solved the equations of motion described by Euler angles [Ref. 74]. However, this finite difference method ignored most warping effects. So, for consistency, for the comparison illustrated in Fig. 16 *only*, the warping term in Equation (3-58) was left out and pure sine torsional mode shapes were used for the modal analysis instead of the mode shapes described in Section 3.3.2.

The results of this comparison of methodologies are presented in Fig. 16 for the $[0_3/90]_S$ wing. Note that because the warping terms have been ignored, the linear natural frequencies at 0 cm do not correspond to those in Table 2.

A minimum of two fore-&-aft modes are required to sufficiently describe the proper trend in first torsional frequency variation; a minimum of three torsion and three fore-&-aft modes are required to sufficiently describe the trend in the second torsional frequency; and a minimum of four torsion and four fore-&-aft modes are required to sufficiently describe the trend in the third torsional frequency. For accuracy, as compared to the exact analysis, three

102

torsion and three fore-&-aft modes seem to be necessary for describing the first and second torsional frequency variations, while five torsion and five fore-&-aft modes seem to be necessary for the third torsional frequency. So, in general, a minimum of three torsion and three fore-&-aft modes are required to accurately predict the first torsion frequency, with an additional torsion and an additional fore-&-aft mode required for each subsequent torsion frequency.

The trends exhibited by both the exact and the modal analyses indicate a softening trend in all the torsional frequencies — a drop of approximately 10% in the first torsion frequency, a drop of approximately 30% in the second torsion frequency, and a drop of approximately 15% in the third torsion frequency. These trends reflect those observed by Minguet [Ref. 74] for specimens of much higher aspect ratio — semi-span AR=18 as opposed to semi-span AR=4 for the current analysis — but show a less marked drop in frequency as tip deflection increases. This less noticeable coupling is likely due to the high stiffness of the fore-&-aft mode which comes from the large chord-to-thickness ratio — in the current investigation, the fore-&-aft stiffness is four orders of magnitude greater than the out-of-plane bending stiffness. The layups are so stiff in the fore-&-aft direction that the v component is very small, and couples only lightly into the lower mode torsional θ motion.

It should also be noted that the exact values are stiffer (i.e. higher in frequency) than those predicted by the modal analysis. This is less noticeable for the first torsion frequency, but more so for the second and third torsion frequencies. — eg. at 20 cm tip deflection the exact and modal analyses for the second torsion frequency

103

differ by approximately 10 Hz, or about 20% of the exact value. This discrepancy is rooted in the two assumptions of the modal analysis. First, the modal analysis is based on an ordering scheme for moderate deflections, whereas the exact analysis is valid for arbitrarily large deflections. Also, the fact that a discrete number of modes is being used in the modal analysis will affect *the final stiffness of the problem.* The first of these two effects is the major contributor to the discrepancy, since with more and more modes the analysis is still converging to frequency values below those of the exact analysis.

From this comparison it can be estimated that the modal analysis is accurate up to "moderate" deflections of about 10% of the span (or about 5 cm), and are not accurate but follow the correct softening trend for "large" deflections, i.e. above 10% of the span. If the second and third torsional modes were directly involved in the flutter analysis, then this discrepancy at large deflections would adversely affect the analysis since the wings are likely to either be diverged or else flutter at high velocity, and thus the typical deflection at flutter would likely be above 10 cm. However, the second and third torsional modes are only really used in the analysis as minor corrections to the lower modes, so the previously mentioned discrepancy is not likely to adversely affect the final flutter analysis.

It should also be noted that for very few fore-&-aft modes the analysis is entirely spurious, for example 3 torsion & 1 fore-&-aft for the 1st torsion frequency, 3 torsion & 2 fore-&-aft for the 2nd torsion frequency, and 3 torsion & 3 fore-&-aft for the 3rd torsion frequency. The analyses using these parameters show a rapid hardening trend instead of a slow softening trend. In these cases, there are

so few fore-&-aft modes that the torsion/torsion nonlinear terms (characterized by R_{mnij} of equation (3-157)) dominate the torsion/fore-&-aft nonlinear terms (characterized by H_{mij} of equation (3-158)).

In terms of application to flutter analysis, it is important that while the second and third torsion frequencies show moderate softening, the first torsion frequency shows little change at all (only 2 Hz change over a range of 40% tip deflection, which is within the error of the linear analysis presented in the previous section), and the modal analysis matches closely the exact analysis. Since the first torsion mode is dominant in both the coalescence of a linear flutter analysis, or the single degree of freedom of a nonlinear flutter analysis, it is clear that the nonlinear geometric effects will have little influence on the flutter solution. While it is true that not only the torsion frequency will also be affected by the nonlinear geometric effects, but also the torsion mode shape, again, because the fore-&-aft stiffness if so large, the fore-&-aft contribution to the altered mode shape is negligible. What little contribution there is from the fore-&-aft mode has essentially no influence on the aerodynamics — the fore-&-aft velocity is so small as compared to the free stream velocity that the dynamic pressure is negligibly affected. There would also be a contribution to the flapping rate by the rotation of the fore-&-aft velocity from the local wing frame into the frame of the free stream — again, this is second order and negligible because it involves the product of two small quantities, the fore-&-aft velocity and the angle from the local twist.

Fig. 17 shows the results of the modal analysis with the warping terms included and demonstrates that the trends remain the same and of the same order of magnitude as for Fig. 16. The second and third torsion frequencies show the same trends of moderate softening. The first torsion frequency exhibits slight hardening instead of slight softening, but the increase of approximately 2 Hz over a 40% tip deflection still remains not significant enough to merit ignoring the geometric nonlinearities in a flutter analysis. It should be noted here the noticeable change between Figs. 16 and 17 in linear frequency from analysis without and with torsional warping terms included. As noted in the previous Section 5.1.2, the warping values β are relatively small, so the first torsion frequency changes only a small amount (approx. 21 Hz to 25 Hz, or 20% change). However, the increased dependence of the higher modes on β, and the strong influence of the discontinuity of the styrofoam fairing near the wing root, cause the changes in the second and third torsional frequencies to be more noticeable (approx. 63 Hz to 79 Hz, or 25% change, and 105 Hz to 145 Hz, or 40% change).

106
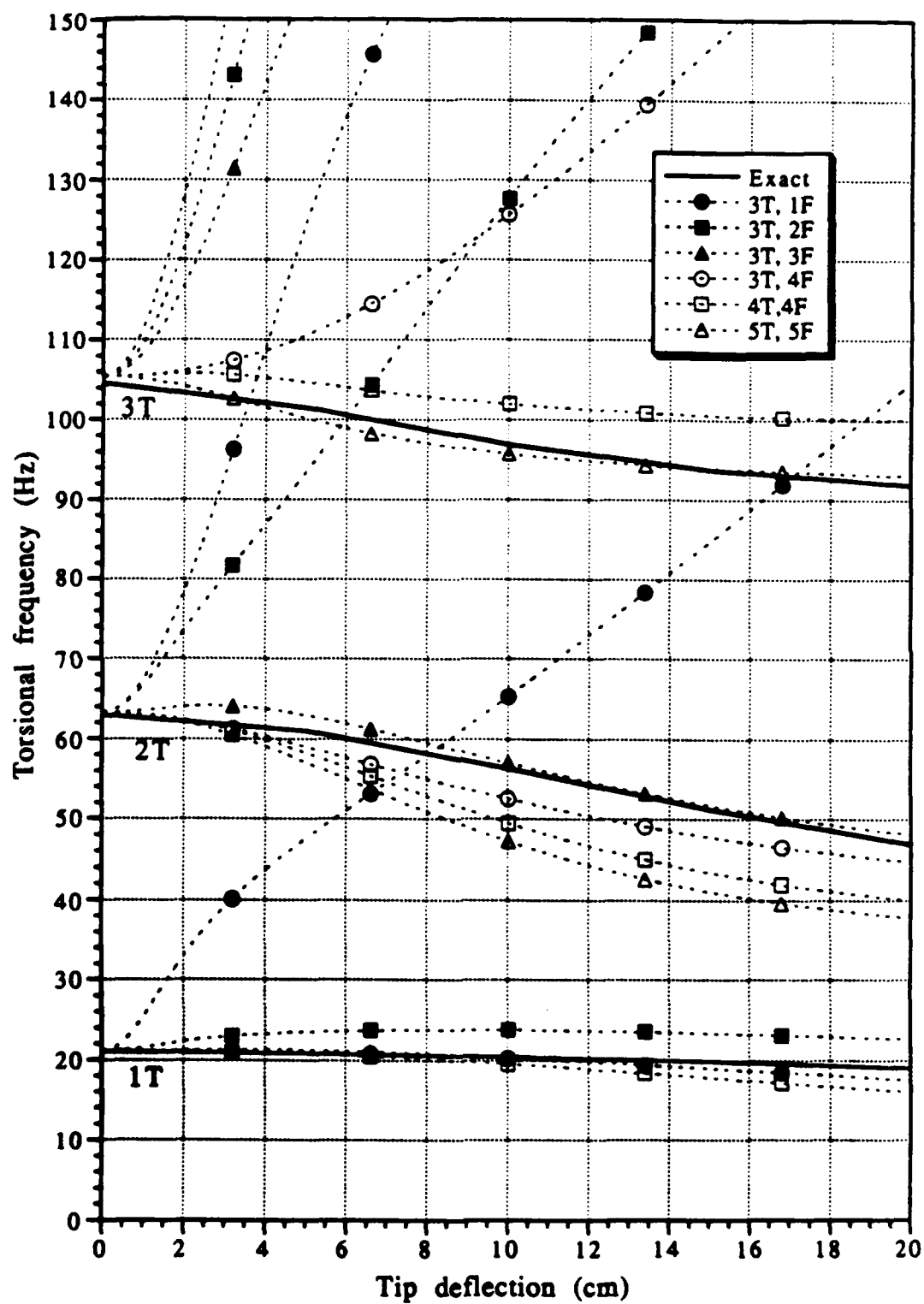
Fig. 16 $[0_3/90]_S$ wing frequency variation, torsional warping terms not included
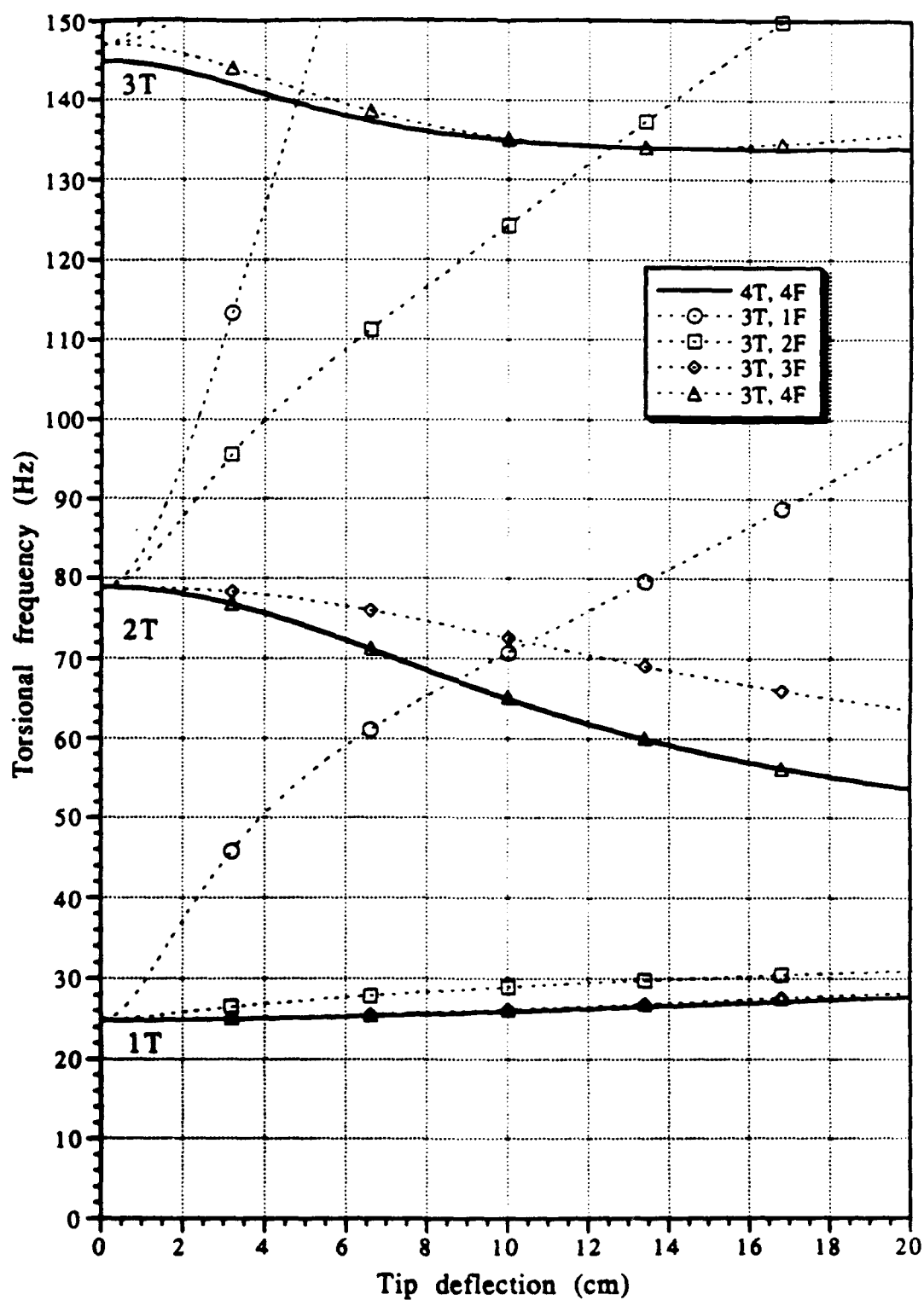
Fig. 17 $[0_3/90]_S$ wing frequency variation, torsional warping terms included

108

5.1.4 Two-Dimensional Aerodynamics

Different analysis methods for two-dimensional coefficient hysteresis for a NACA 0012 airfoil in low Reynolds number flow are compared against one another in Figs. 18 and 19. The analysis methods that are compared are constant coefficient harmonic balance, non-constant coefficient harmonic balance, and fourth-order Runge-Kutta time marching. The purpose of comparing these analysis methods is to determine first, whether non-constant coefficients are required instead of constant coefficients, and second, whether a harmonic analysis is sufficient to accurately describe the hysteresis loop as compared against a method that is presumably an exact solution to the equations of motion.

Fig. 18, for a moderate amplitude of oscillation of 4°, over a range of reduced frequencies from 0.10 to 0.25, indicates that the non-constant coefficient analysis compares favorably against the exact Runge-Kutta analysis. This figure also indicates the major deficiency of the constant coefficient analysis: since the coefficients are constant, there is no nonlinear coupling between the mean of the force coefficient (C_{L_0} in Fig. 18) and its other harmonic components, and therefore there is no dependence of the mean on the reduced frequency. However, it is clear from the non-constant and Runge-Kutta analyses that the mean of the force coefficient is in fact strongly influenced by the reduced frequency.

Table 3 indicates more clearly the appropriate trends for the first harmonics, which are the harmonics that are likely to be most dominant in a full flutter analysis. For the moderate amplitude of

oscillation of A1=4°, both the Runge-Kutta and non-constant analyses indicate a decrease in mean force coefficient as reduced frequency increases (of about 20% from k=.10 to k=.25), while the constant coefficient analysis shows no change. This is because the constant coefficient analysis has no coupling between the harmonics and the mean, so the mean value remains unchanged as long as the oscillation amplitude, A1, remains unchanged. While the non-constant analysis doesn't exactly match the exact Runge-Kutta analysis, and drops more quickly with reduced frequency, the values are within 10% error relative to the Runge-Kutta analysis.

The in-phase, sine components follow the same trend for the Runge-Kutta and non-constant analyses and are relatively close in magnitude — there is a slight drop in sine values from k=.10 to k=.15 then an approximate doubling from k=.15 to k=.25. The Runge-Kutta and non-constant analyses are within 35% of one another for the sine components. However, the constant coefficient analysis rises continuously from k=.10 to k=.15 to k=.25. It would be expected that the sine values would be fairly close for all analyses, since the in-phase component is governed principally by the linear static slope.

The out-of-phase, cosine components match well for the Runge-Kutta and non-constant analyses (within 25% of one another), showing an increasing trend with reduced frequency, while again the slope for the non-constant analysis is greater than that for the Runge-Kutta analysis. However, the constant coefficient analysis shows a different trend, namely a slight rise from k=.10 to k=.15 then a large drop of approximately 25% at k=.25 — this has serious repercussions for a flutter analysis since the out-of-phase term is an

important indication of the work being performed by the fluid flow. This 2-dimensional aerodynamic analysis indicates that the constant coefficient analysis is probably inappropriate for producing accurate flutter results.

Fig. 19 shows that the non-constant coefficient analysis breaks down for very large amplitudes of oscillation, in this case $A1=10^o$. In general, the non-constant coefficient analysis is only valid to amplitudes of oscillation in the range of 5° or 6°. Table 3 shows that the mean and sine components are still moderately close for the Runge-Kutta and non-constant analyses (within 10-20% of one another), however the out-of-phase cosine component — while still exhibiting the same increasing trend for both methods— remains about 300% off for all values of reduced frequency. This is another indication that even when only considering the first harmonics, the harmonic analysis begins to break down for large amplitudes of oscillation. Again, the constant coefficient analysis indicates an unchanging mean value (since there is no nonlinear coupling between the harmonics), while the sine and cosine components are even further off the mark at $A1=10^o$ than for the moderate amplitude of $A1=4^o$.

At first glance, from looking at the values for k=.10 and k=.15, one might conclude that the mean values decrease as amplitude of oscillation increases, but this is not the case. Instead, the trend changes with increasing amplitude of oscillation — that is, the mean decreases for increasing reduced frequency at $A1=4^o$, but the mean *increases* for increasing reduced frequency at $A1=10^o$. So, for increasing amplitude of oscillation, the mean drops at low reduced frequencies and rises at high reduced frequencies, and there is some

point between k=.15 and k=.25 for which it does not change at all. A similar change is also noticed for the sine component — the sine component generally increases for increasing reduced frequency at A1=4°, but the sine component decreases for increasing reduced frequency at A1=10°. It is difficult to attribute physical interpretations to these trends because of the high nonlinearity of the formulation.

Overall, with the two harmonics also taken into consideration, the analytic hysteresis loops look odd at an oscillation amplitude of A1=10° in Fig. 19. The loops appear to not have enough stall delay, drop too low and too quickly into stall after the stall delay, and return too suddenly from nonlinear stall to linear behavior in returning below the stall angle. This odd behavior is likely due to the fact that the fixed-time stall delay, $\Delta\tau$, has been smeared over the entire cycle, making the apparent delays within specific portions of the loop seem incorrect.

Fig. 20 is an example of experimental data from Ref. 5. While this data does not match the analysis — $5x10^5$ Reynolds number as compared to $2x10^5$ for the analysis in Figs. 18 & 19 — it is worthy to note that the trends in the experiment are similar to that of the analysis at moderate amplitudes of oscillation. The two harmonic Fourier components of the data are also presented in Fig. 20.

It was also attempted to compare the analysis against results produced by the Upwind Approximate Factorization Navier-Stokes (UPWAFNS, *a.k.a. CFL2D*) computational fluid dynamics code [Ref. 30]. However, because of the large amplitudes being considered here, the CFD analysis proved to be unsuccessful in satisfactorily reproducing the experimental results of Ref. 5 presented in Fig. 18.

112

	mean			sin(kτ)			cos(kτ)		
	RK	NC	CC	RK	NC	CC	RK	NC	CC
A0=10 A1=4 k=.10	.699	.740	.765	.150	.098	.185	.114	.098	.218
A0=10 A1=4 k=.15	.690	.711	.765	.137	.091	.284	.153	.196	.236
A0=10 A1=4 k=.25	.578	.520	.765	.222	.259	.339	.229	.242	.174
A0=10 A1=10 k=.10	.512	.509	.567	.347	.381	.223	.010	-.004	.142
A0=10 A1=10 k=.15	.519	.572	.567	.293	.257	.169	.023	.062	.321
A0=10 A1=10 k=.25	.644	.886	.567	.125	-.089	.317	.106	.329	.740

RK = Runge-Kutta analysis
NC = non-constant coefficient analysis
CC = constant coefficient analysis

Table 3. 1st harmonic components of 2-dimensional aerodynamic analysis

113
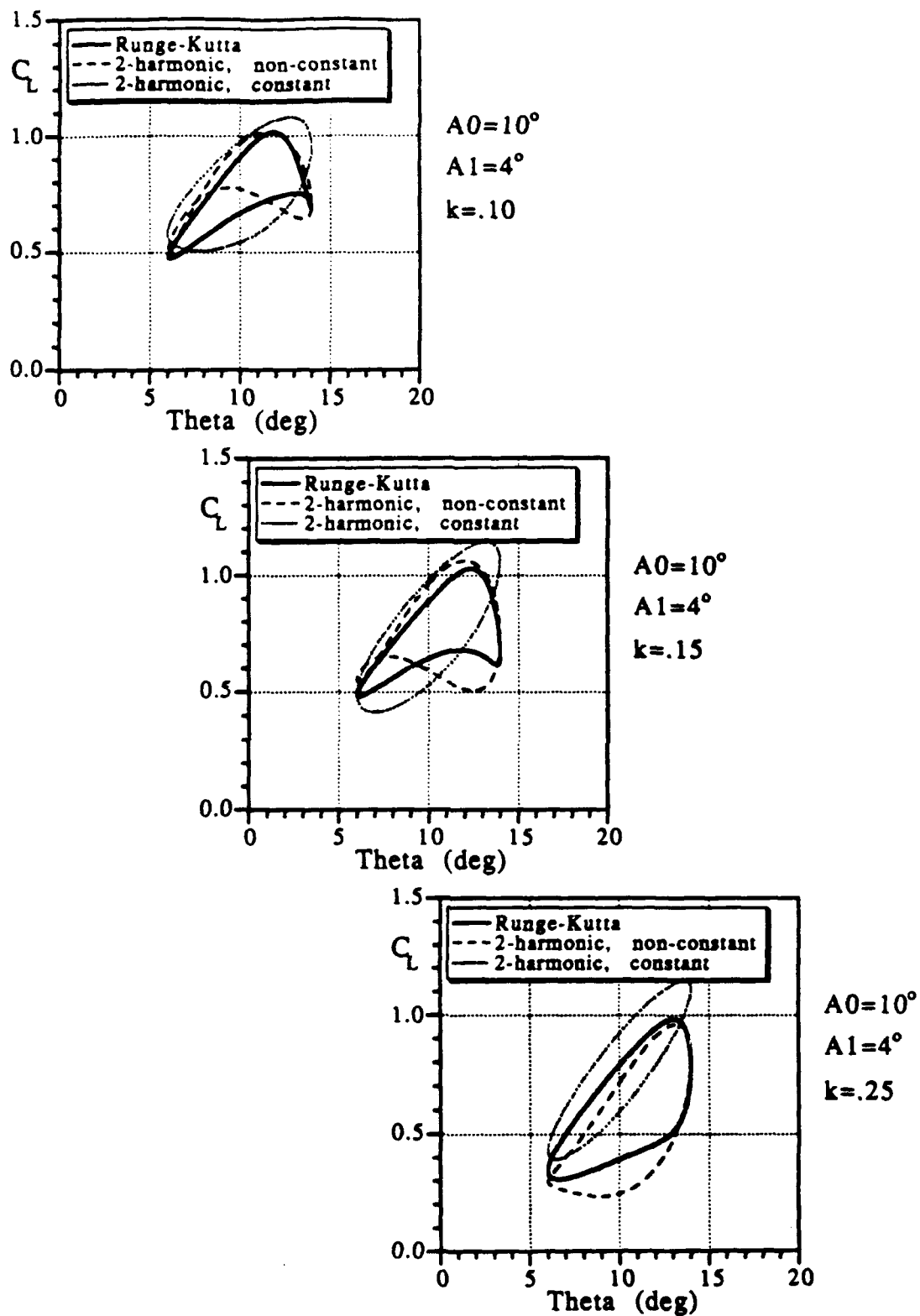
Fig. 18 2-dimensional lift coefficient hysteresis loops, analysis for NACA 0012 airfoil, Re=.2E6
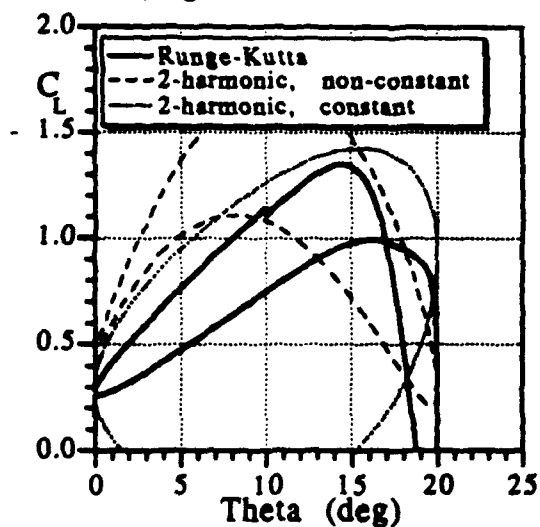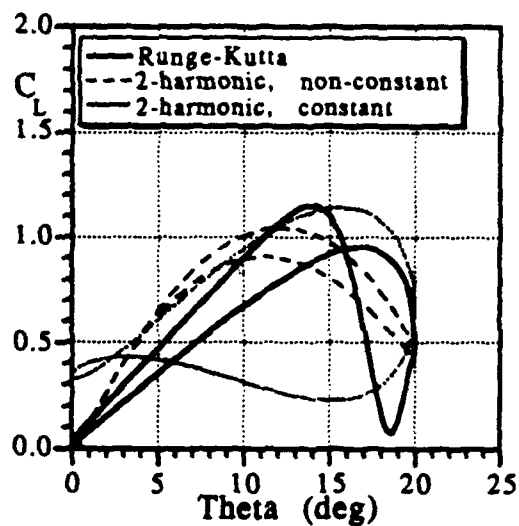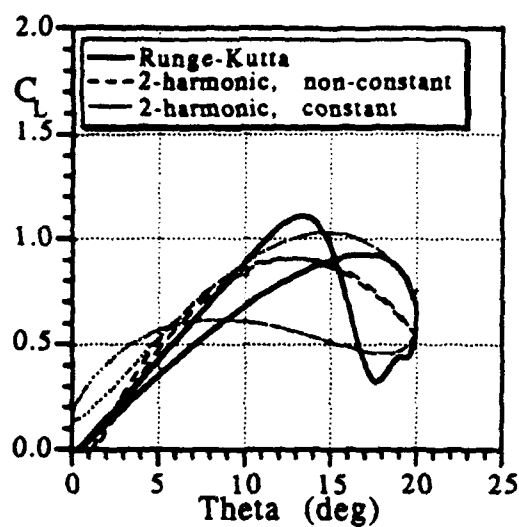
Fig. 19 2-dimensional lift coefficient hysteresis loops, analysis for NACA 0012 airfoil, Re=.2E6

115

AO=9.82°
A1=9.90°
k=.104

AO=9.80°
A1=9.92°
k=.151

AO=9.78°
A1=9.93°
k=.253
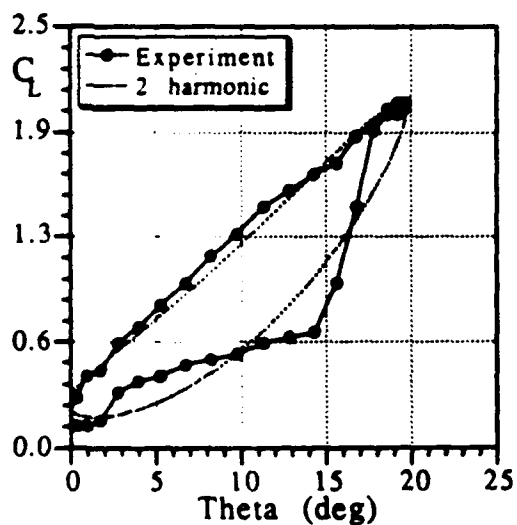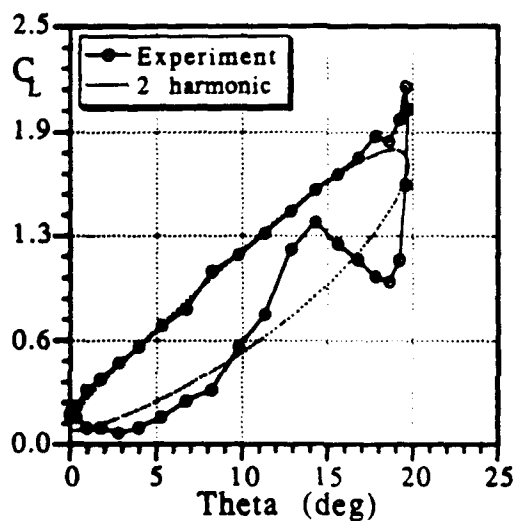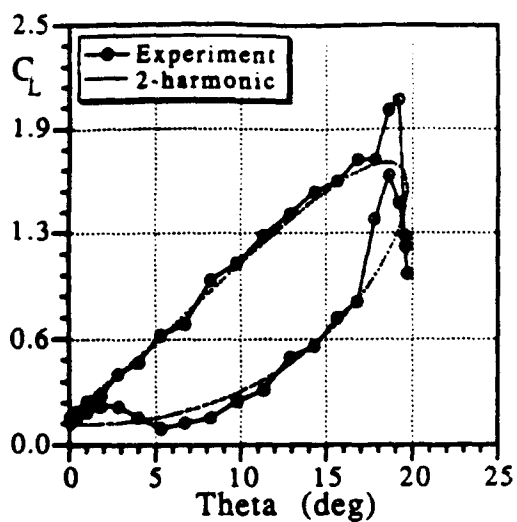
Fig. 20 2-dimensional lift coefficient hysteresis loops, experiment for NACA 0012 airfoil, Re=.5E6

5.2 Flutter Analysis

5.2.1 Linear U-g Analysis

The analytic results for classical, linear divergence are presented in Table 4 for the laminates of interest using the three-dimensional, linear aerodynamics described in Section 3.5.1. The U-g plots from which these values are generated, using three bending mode shapes and three torsional mode shapes, are shown in Figs. 21 to 23. As expected, the $[0_3/90]_S$ wing has a torsional flutter velocity very near its divergence velocity, the $[+15_2/0_2]_S$ wing has a torsional flutter velocity just above that of the $[0_3/90]_S$ wing, and the $[-15_2/0_2]_S$ wing has a divergence velocity well below its torsional flutter velocity.

For the $[0_3/90]_S$ layup, the linear flutter velocity, V_F, is approximately equal to the linear divergence velocity, V_D, because there is no bending-torsion coupling and the elastic axis is at the midchord — both are near 28 m/s which is just below the experimental wind tunnel limit. The flutter frequency, ω_F, is the typical coalescence frequency of a linear flutter analysis and, as seen from the U-g plot in Fig. 21, results from the drop of the first torsional frequency ω_{1T} from 24.6 Hz to 11.9 Hz, while the first bending frequency ω_{1B} drops from 4.3 Hz to 0 Hz (i.e. divergence). The combination of the flutter velocity and the flutter frequency yield a reduced frequency of approximately 0.15, indicating that any nonlinear analysis applied at this point is within the valid range discussed in the previous section. The divergence velocity is in fact slightly below the flutter velocity, so in a real flutter situation one would expect stalling at the

point of flutter initiation, even at root angle of attack of $\alpha_R = 0°$. However, one would also expect this stalling to be light, since the nonlinear effects will just be beginning to come into play so as to hinder the exponential growth in deflection due to divergence. It is not possible to predict the dependence of the flutter characteristics on an increasing root angle of attack α_R since the aerodynamics are stalled, and hence will likely affect both the flutter velocity and the flutter frequency.

For the $[+15_2/0_2]_S$ layup the linear divergence velocity is much greater than the linear flutter velocity due to the "negative" bending-torsion coupling of the layup — that is, because positive bending induces negative twist. As noted in Section 5.1.2, the flutter characteristics have more to do with the natural mode shapes than the natural frequencies, since the natural frequencies remain relatively unchanged for all the layups. The behavior of the first torsion mode is almost identical to that of the $[0_3/90]_S$ layup: as shown in Fig. 22, the first torsion frequency ω_{1T} drops from 23.5 Hz to 11.6 Hz by the time the damping ratio crosses the zero axis at 26.9 m/s. However, the first bending frequency continues to rise from 3.9 Hz, and the damping ratio in bending continues to decrease, never crossing the zero axis. However, this might be simply due to the fact that the U-g analysis has not been carried out to a high enough velocity — there might in fact be a velocity high enough that the bending-torsion coupling becomes too weak to counteract the positive twist created by the increasing moment induced by the associated increasing dynamic pressure. Therefore, the notation of "∞" in Table 4 is only meant as an indication that the divergence velocity might be very

118

large, and is certainly out of the range of the velocity plotted in Fig. 22 and out of the range of the actual wind tunnel experiment.

In a real flutter situation with the $[+15_2/0_2]_S$ layup, stalling would again be expected, but not for the same reason as divergence as for the $[0_3/90]_S$ layup. Instead, it would be expected that the amplitude of oscillation would grow exponentially until it reaches the stalling regime, at which point the stalling would hinder further growth and induce limit cycles. Therefore, certain characteristics can be predicted within the range of root angle of attack α_R below the static stall angle. First, it can be expected that the flutter velocity and frequency will remain relatively unchanged since the flutter response will be governed by linear aerodynamics (with slight corrections to rotate the aerodynamic forces through α_R into the local wing coordinates), and the nonlinear aerodynamics will only act as a limiting factor. It is in this limiting factor that the characteristics will change with increasing root angle of attack: as α_R increases, smaller and smaller amplitudes of oscillation will be required to reach the stalling regime and induce limit cycles. Therefore, as the root angle of attack is increased, it is expected that the flutter velocity and frequency will remain unchanged while the amplitude of oscillation decreases. This trend seems counter-intuitive at first — one generally expects increasing response (whether steady deflection or unsteady oscillation) with increasing load due to an increased root angle of attack.

For the $[-15_2/0_2]_S$ layup the linear divergence velocity is much below the linear flutter velocity because of the "positive" bending-torsion coupling — that is, because positive bending induces positive

119

twist. In a general physical sense, divergence is induced because an aerodynamic load causes a positive twist, which induces further aerodynamic load, which causes further twist, and so on, in a cascading effect. With the "positive" bending-torsion coupling, the increasing aerodynamic load also induces an increased bending, which also induces further twist, and thus "aggravates" the divergence effect. That is, because of the bending-torsion coupling, the velocity at which this effect continues to cascade (i.e. grow exponentially) instead of converging to a steady deflection is lower than for a layup with no bending-torsion coupling, such as the $[0_3/90]_S$ layup. The U-g diagram for the $[-15_2/0_2]_S$ layup, shown in Fig. 23, indicates a different behavior for the torsional mode than the two previously discussed layups. The behavior shows no sharp rise in damping ratio, and no sharp drop in the torsion frequency ω_{1T} just before the onset of flutter — as is the case for the $[0_3/90]_S$ and $[+15_2/0_2]_S$ layups in Figs. 21 and 22 — but is instead more gradual.

In a real aeroelastic situation, these results would indicate that the wing would likely experience divergence, whose growth would probably be limited by nonlinear aerodynamic effects. Whether such a growth would trigger any kind of oscillatory response is not predictable by the U-g analysis. It might also be possible to achieve flutter at the higher, linear flutter velocity, but this is unlikely. First, because of divergence, the wing might possibly break at such a high velocity. Second, again because of divergence, at the higher, linear flutter velocity the aerodynamics will be in deep stall, and will be very different from those used to predict the linear flutter velocity in the first place.

120

Therefore, the U-g method has various degrees of applicability in predicting flutter characteristics for a range of bending-torsion coupling. For the $[0_3/90]_S$ layup, the U-g method is likely to accurately predict the correct flutter velocity and frequency at root angle of attack $\alpha_R = 0$, but one can only estimate trends for increasing α_R. For the $[+15_2/0_2]_S$ layup, the U-g method will accurately predict the flutter velocity and frequency characteristics over a wide range of root angles of attack up to the static stall angle, although it can only be used to estimate the trend in limit cycle amplitudes over that range. For the $[-15_2/0_2]_S$ layup, the U-g method will likely predict the correct onset of an aerodynamic instability at the divergence velocity, but what type of instability — whether divergence or oscillatory — is uncertain. As with the $[0_3/90]_S$ layup, it is difficult to predict the behavior for increasing root angle of attack α_R.

NACA 0012 WINGS U-g ANALYSIS RESULTS		
V_D, Divergence Velocity (m/s)	V_F, Flutter Velocity (m/s)	ω_F, Flutter Frequency (Hz)
$[0_3/90]_S$ 28.20	28.27	11.86
$[+15_2/0_2]_S$ ∞	26.89	11.64
$[-15_2/0_2]_S$ 18.04	35.60	13.34

Table 4. Linear divergence and flutter characteristics
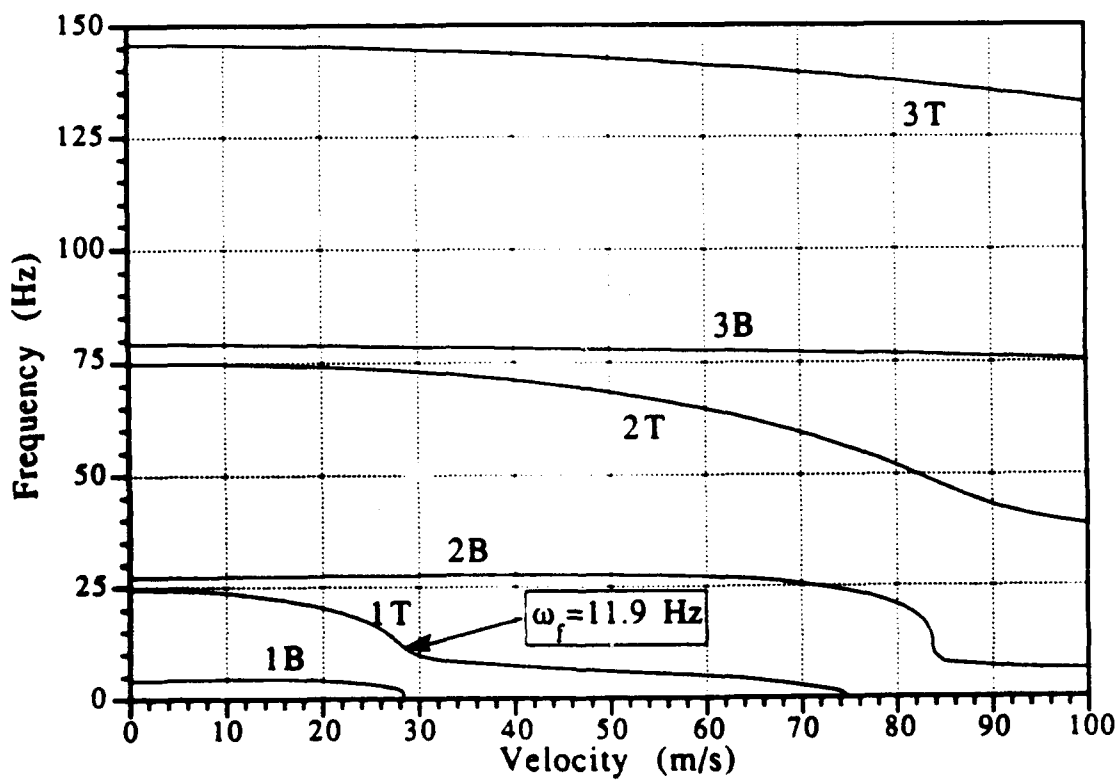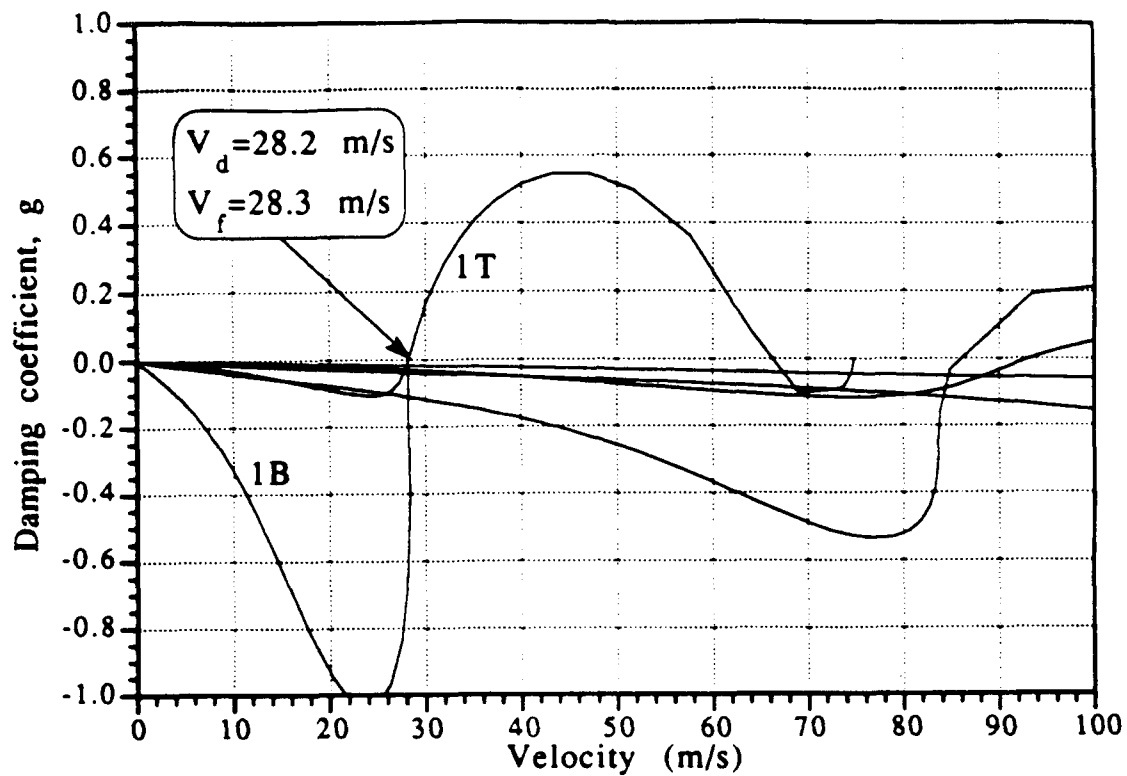
Fig. 21 $[0_3/90]_S$ **U-g diagram**
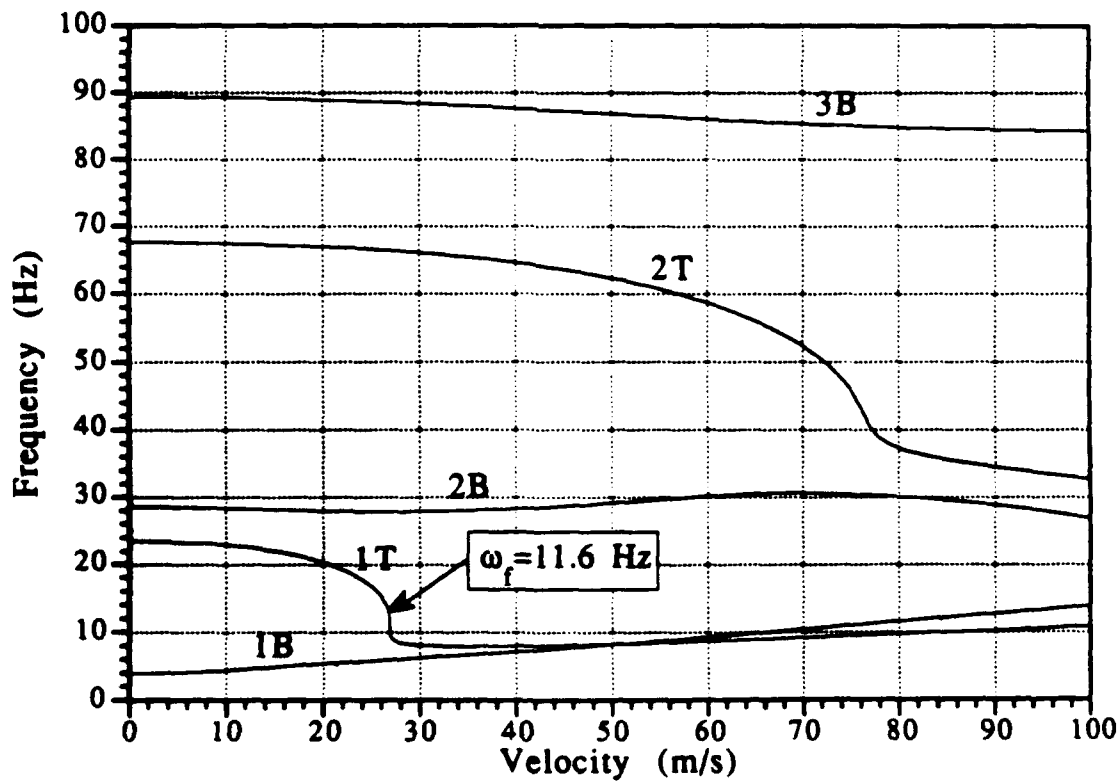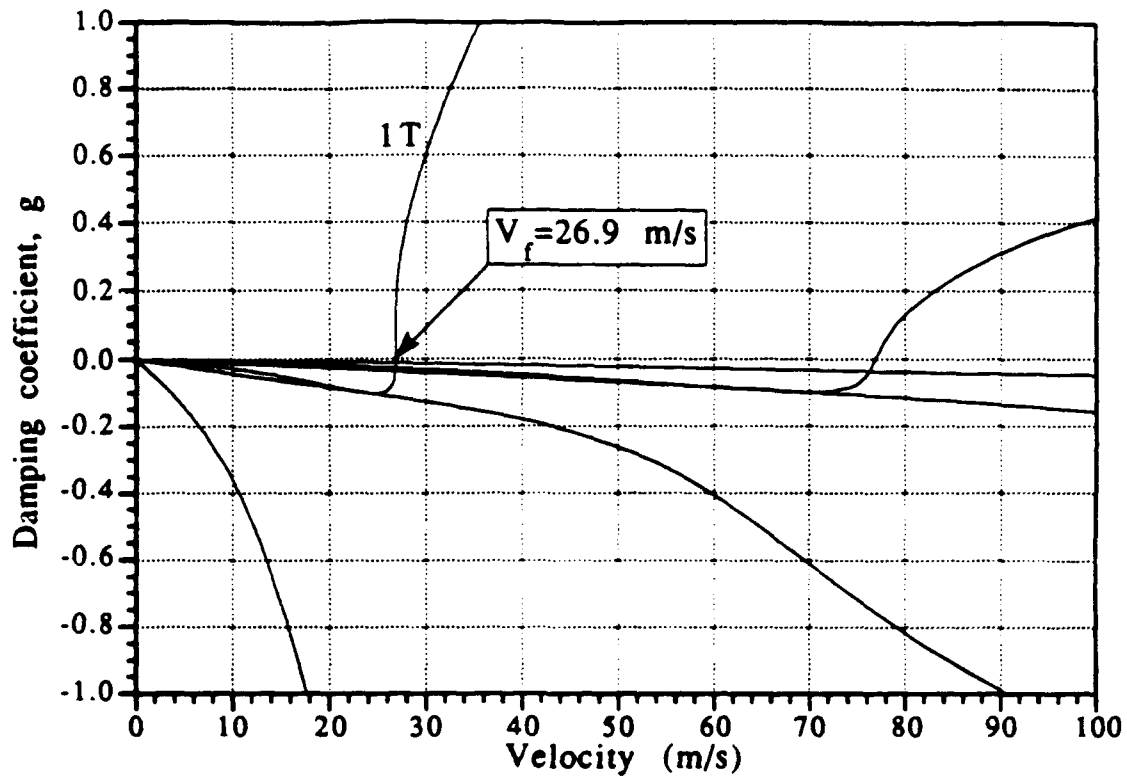
Fig. 22 $[+15_2/0_2]_S$ U-g diagram
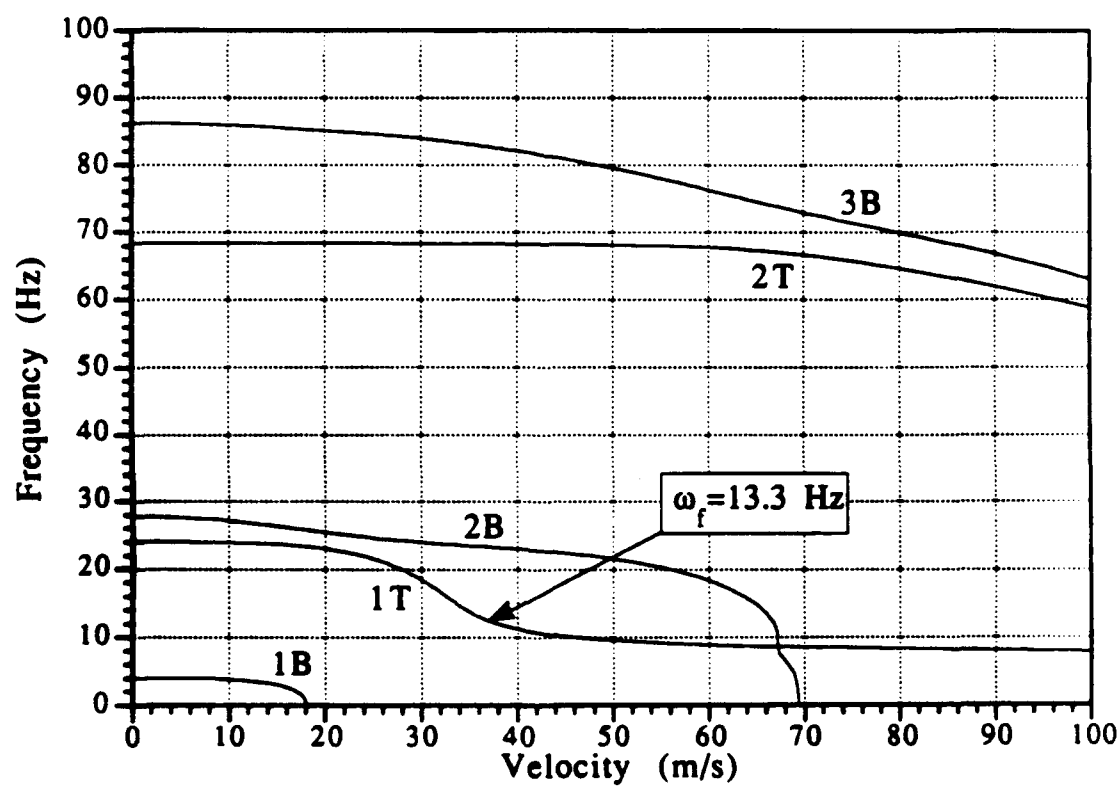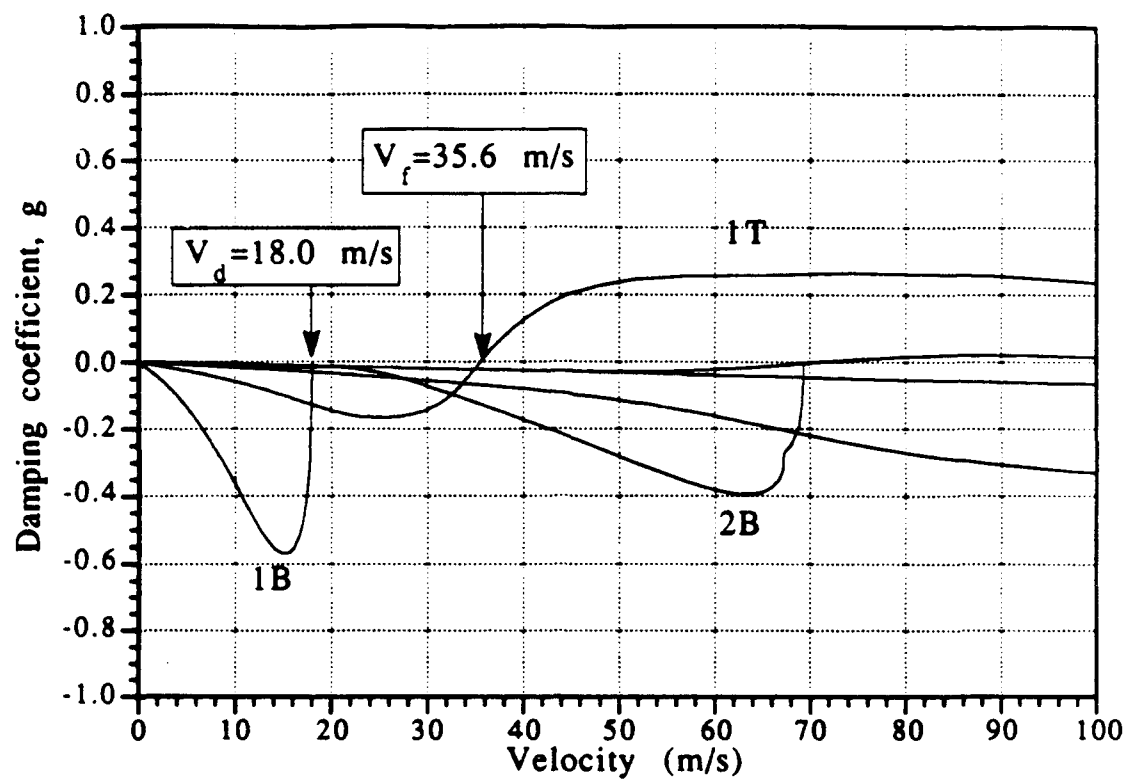
Fig. 23 $[-15_2/0_2]_S$ U-g diagram

5.2.2 Flutter Boundaries

The experimental and analytic flutter boundaries (i.e. for very small amplitude oscillation) are presented in Figs. 24 to 26. Each graph demonstrates some of the expected trends for each of the $[0_3/90]_S$, $[+15_2/0_2]_S$, and $[-15_2/0_2]_S$ layups that were predicted by the U-g analysis, but quantifies these trends in a way that the U-g analysis could not.

Fig. 24 for the $[0_3/90]_S$ laminate starts at the linear flutter velocity but immediately begins to exhibit nonlinear behavior because the linear flutter velocity is so close to the divergence velocity, as would be expected from the U-g analysis. That is, the divergence and exponential growth of flutter into the stalling regime only limit the growth to limit cycles, but do not significantly alter the linear results that could be derived by the U-g analysis, otherwise the nonlinear results would be further from the linear results at root angle of attack $\alpha_R = 0°$. An increase in the root angle of attack α_R causes the flutter velocity to drop and the flutter motion to become more purely torsional (denoted by a frequency closer to the first torsion free vibration frequency and a decrease in the bending amplitude).

For values of the root angle of attack up to the static stall angle, this behavior is governed by "light" stalling, i.e. where the major portion of the wing is oscillating across the static stall angle, back and forth between the stalled and unstalled regions. The flutter velocity drops smoothly and slowly as the root angle of attack increases, by about 1 m/s per 1° increase in α_R — as the root angle of attack

126

increases, the distributed load consequently increases, and the wing would twist further into stall, except that the velocity decreases so as to decrease the dynamic pressure, thus counteracting the positive twist, and keeps the wing only in light stalling. Also, as the flutter velocity drops, the flutter frequency consequently rises toward the first torsional natural frequency — again, because the stalling effects only induce the limit cycles and do not strongly affect the linear aerodynamics, this is the same as travelling backward along first torsion branch of the frequency plot derived by the U-g analysis.

After the root angle of attack reaches the static stall angle, the flutter behavior begins to be governed by "deep" stall, that is, the changes in characteristics of the vortex shedding as the flow gets pushed further in the stall regime — characterized mathematically by the parameters a_1, r_1, and e_1 in equations (3-94) to (3-96). The flutter velocity continues to decrease smoothly, and the flutter frequency continues to increase smoothly toward the first torsional natural frequency, indicating a strong dependence on deep stall characteristics. If the deep stall characteristics were in fact weak, then the flutter velocity and flutter frequency would show little change past the static stall angle since both the aerodynamic linear characteristics and the nonlinear stalling characteristics would change very little (C_L levels off past stall, while C_M continues to drop, but only slowly — see Appendix C). These trends indicate that in designing an airfoil for flutter purposes, it might be desirable to do so such that the deep stalling characteristics change as little as possible from the light stalling characteristics, so that the flutter velocity in deep stall remains as high as possible. This objective, however, might not

127

prove possible since, as noted in early dynamic stall studies [Refs. 1-5], the characteristics of deep stall tend to be independent of airfoil geometry.

It should also be noted that experimentally, at root angle of attack $\alpha_R = 1°$, *simultaneous* bending and torsion flutter were observed at the divergence/flutter speed (although only the experimental, torsional flutter speed is plotted on Fig. 24). The wing would "flap" at large amplitudes of oscillation in bending then, intermittently, would cease flapping and instead oscillate in a torsional manner. This observation seems to indicate the strong coupling of both divergence and flutter in the linear regime.

Fig. 25 for the $[+15_2/0_2]_S$ laminate shows a more extended range of linear aerodynamic behavior as would be expected from the linear U-g analysis (because the divergence velocity is very high and the tip twist is negative) and a very sharp change in the flutter behavior once it goes into the nonlinear stall region. The linear region of the behavior is valid up until the root angle of attack α_R reaches the static stall angle. This behavior is because of the bending-torsion coupling: as α_R increases the distributed load increases, causing increased deflection, and thus inducing negative twist which counteracts the positive twist from the increased distributed moment. In the case of the $[+15_2/0_2]_S$ layup, this effect is large enough so as to actually produce a negative twist, but it would be possible that with a weaker negative bending-torsion coupling the effect would only cause a small, but still positive, twist.

The analysis shows an almost linear trend up to the static stall angle — the only deviation is due to the rotation of the aerodynamic

128

loads from the free stream coordinates into the rotated, local wing coordinates. The analysis also remains almost linear for a short range past the static stall angle, up to about $\alpha_R = 13^\circ$ — the negative twist keeps the majority of the wing unstalled so that while the root of the wing is stalled, it has little effect on the overall aerodynamics governing the flutter behavior.

Once past the linear region, the flutter behavior goes into deep stall very quickly. This is because by the time the tip finally reaches stall, the root must already be in deep stall because of the negative twist. As the root angle of attack is then further increased, the twist remains essentially unchanged — since the force coefficient curves level off, no more distributed aerodynamic load is generated — so there is no longer any increasing negative twist to counteract the increasing positive angle of attack, and the wing quickly goes into deep stall. In other words, the transition through light stall is unlike that of the $[0_3/90]_S$ layup, and is very short and very sudden. As the light stalling characteristics become less dominant (eg. for an airfoil geometry for which r_o might be smaller), it would be expected that this quick drop would become even more sudden, since the behavior would lock into the deep stall characteristics more quickly. Therefore, a wing with the type of bending-torsion coupling as the $[+15_2/0_2]_S$ layup is beneficial in terms of divergence, but the drop in flutter velocity due to stalling might prove to be sudden and unexpected. For example, a brief change in perceived angle of attack of only 3° — eg. from $\alpha_R = 14^\circ$ to $\alpha_R = 17^\circ$ — might drop the flutter velocity by half its value, while such a small change in angle of attack for

the $[0_3/90]_S$ layup would only induce a moderate change in flutter velocity.

The experimental flutter velocity values for the $[+15_2/0_2]_S$ layup show a smoother trend from linear behavior to nonlinear, stalled behavior than does the analytic prediction. Most noticeably, the experimental flutter velocities drop by 4 m/s between root angles of attack $\alpha_R = 1^o$ and $\alpha_R = 11^o$. There are several possible explanations for this discrepancy. First, there might be unmodeled structural nonlinearities that are unaccounted for in the current analysis. The experimental drop in flutter velocity indicates that there might be an additional softening trend — this might be accounted for by additional geometric nonlinearities, or by cubic stiffening. However, neither of these possibilities seems likely since the inclusion of geometric nonlinearities did not produce this softening trend in the linear aerodynamic region, and cubic stiffening would produce a hardening effect instead of a softening effect. Second, the aerodynamics might not be totally linear just below the static stall angle. In fact, some previous investigations use a parabolic drop just before the static stall angle (see Appendix C), while the current investigation overpredicts the lift coefficient near the static stall angle. This consideration would have the same effect as the light stalling for the $[0_3/90]_S$ layup, and would more smoothly decrease the flutter velocity. This is likely to account for a large part of the discrepancy, but would make the analysis more difficult since it would require more describing regions and higher order approximations than the current analysis. Third, 3-dimensional spanwise aerodynamics effects might also affect the flutter calculation. While

130

the spanwise drop used to taper the aerodynamic load as it reaches the tip (see Appendix C) is fairly accurate for no twist, it tends to be less accurate for either negative or positive twist (see Landsberger [Ref. 92] for comparison of currently used strip theory against 3-dimensional lifting line theory). This effect might also account for some of the discrepancy, though probably very little since the linear coefficients of the approximated Theodorsen function still remain unchanged at each spanwise location.

The discrepancy in frequency in Fig. 25 is easier to account for. First, it is difficult to begin with to accurately get the flutter frequency for frequency coalescence. As can be seen in Fig. 22, the flutter frequency changes very quickly in the range of the flutter velocity, so that any slight structural damping, which might move the zero axis of the damping coefficient and hence slightly alter the flutter velocity, will consequently strongly affect the flutter frequency. Second, the first bending and first torsion free vibration frequencies of the $[+15_2/0_2]_S$ (as listed in Table 2) are in the range of 1 Hz off in comparing experiment to analysis, so it can only be expected for a frequency coalescence phenomenon that the results will show error in the same order of magnitude.

Fig. 26 for the $[-15_2/0_2]_S$ laminate indicates a much different trend where the flutter is characterized by a low, first-bending frequency and immediate nonlinear, bending stall flutter in the range of the divergence velocity — there is no portion of the flutter graph here which could have been predicted by a linear analysis. As with the $[0_3/90]_S$ layup, the behavior seems to be governed by light stall dynamics for a root angle of attack up to the static stall angle. That

131

is, the flutter behavior is triggered only at the very onset of divergence, where the major part of the wing is just starting to straddle the static stall angle, instead of at a higher velocity, where the wing would be twisted even further into stall because of divergence. Again, as with the $[0_3/90]_S$ layup, an increase in the root angle of attack induces a smooth decrease in the flutter velocity of just less than 1 m/s per 1° increase in α_R, as if to keep the governing behavior just bordering the stall regime, neither fully entering either the fully linear or the fully nonlinear, stalled regions. The analytic behavior past a root angle of attack equal to the static stall angle, i.e. in deep stall, also follows the same trend in flutter velocity as the $[0_3/90]_S$ layup, namely that the deep stall characteristics are strong enough so that the flutter velocity continues to decrease at approximately the same rate as for light stall. The experimental behavior for the $[-15_2/0_2]_S$ layup is very different in deep stall than for the $[0_3/90]_S$ layup — perhaps, unlike the $[0_3/90]_S$ layup, the aerodynamics do level off, and the flutter velocity remains relatively unchanged. However, the deep stall data is represented by only one data point at $\alpha_R=15^o$, so it is uncertain whether the discrepancy is due to spurious experimental data or poor analysis.

The trend in flutter frequency for the $[-15_2/0_2]_S$ layup differs quite markedly from that of the $[0_3/90]_S$ layup. Instead of starting at the linear frequency coalescence value, it starts at just below 2 Hz, that is somewhere between 0 Hz and the first bending natural frequency of 4.0 Hz. The effect is likely analogous to that of the $[0_3/90]_S$ layup: the nonlinear stalling features do not play a significant role in changing the 'inear aerodynamics, but instead govern the

132

limit cycles and hence determine at which point on the U-g diagram the solution shifts in accordance with an increasing root angle of attack. So, near the divergence velocity, the flutter behavior is governed by the first bending mode (which has the lowest associated damping ratio as seen in Fig. 23) and as the root angle of attack α_R is increased, thus decreasing the flutter velocity, the flutter frequency consequently increases toward the first bending natural frequency, as if following the first bending branch of the frequency plot for the U-g analysis. The only portion of Fig. 26 that could directly have been predicted by the U-g analysis is the velocity at which the aeroelastic instability first occurs. The U-g analysis would have predicted divergence, but could not have predicted the possible oscillatory nature of the instability, its frequency, or the ensuing trend with increasing root angle of attack — these would require the nonlinear, stalled analysis.

All three figures indicate that as the root angle of attack α_R is increased, the flutter velocity — whether bending or torsional — decreases, and the flutter frequency tends toward the associated linear, natural frequency. The parameter of bending-torsion coupling determines whether the flutter frequency will start as coalescence of the torsion with the bending mode, or will start near the first bending frequency, and whether the decrease in flutter velocity with root angle of attack will be smooth or sudden. The previous study [Refs. 94 and 95] was unable to experimentally investigate the phenomenon of sudden transition from linear flutter to nonlinear, stalled flutter because of the velocity limitations of the wind tunnel.

133
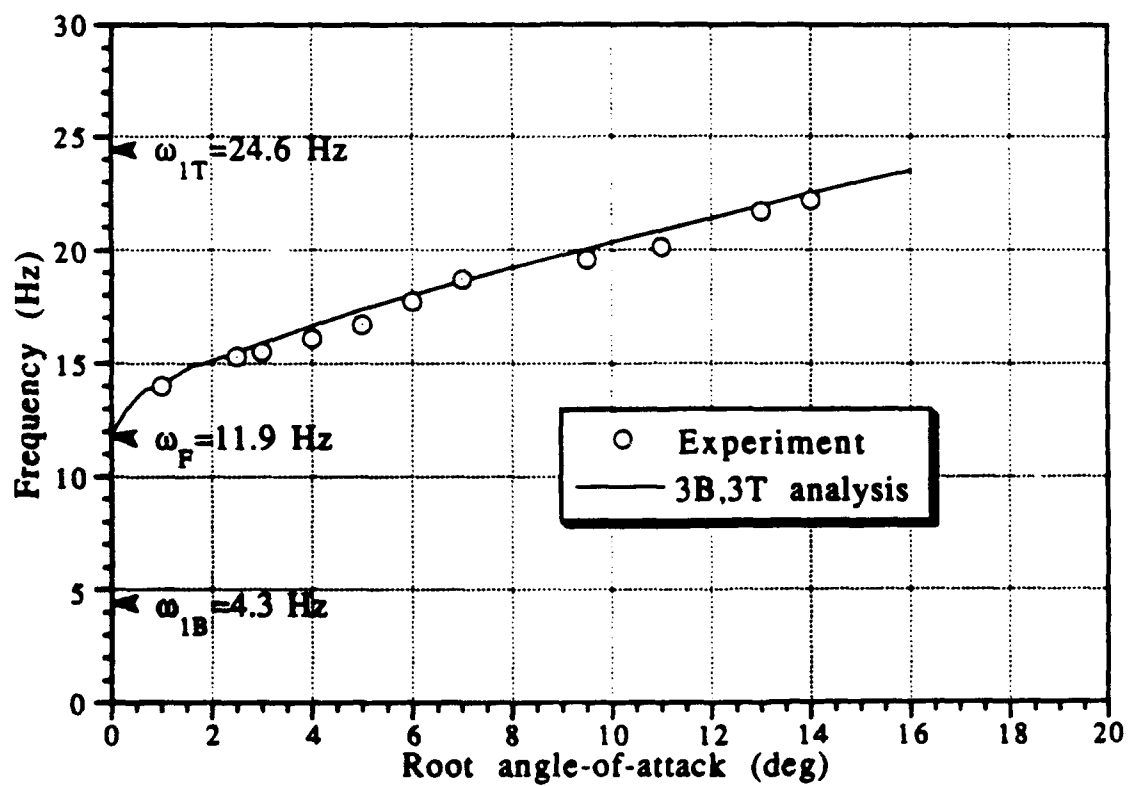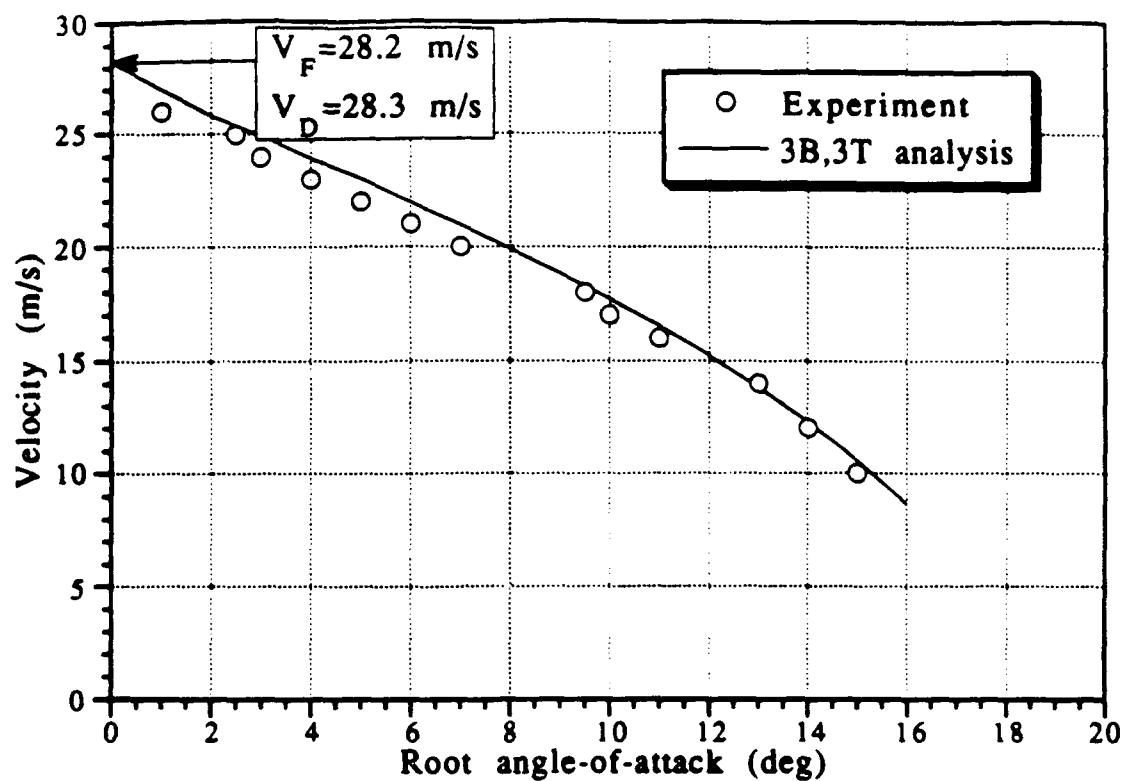
Fig. 24 $[0_3/90]_S$ flutter boundary and frequency variation

134
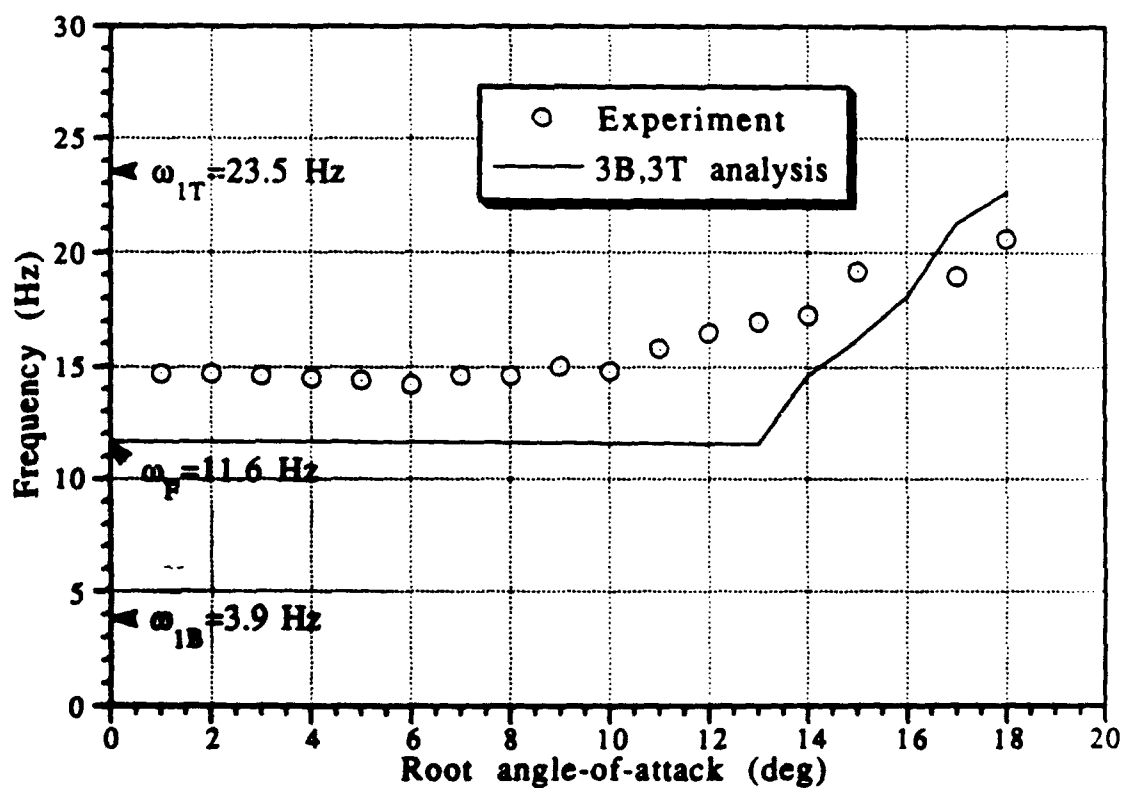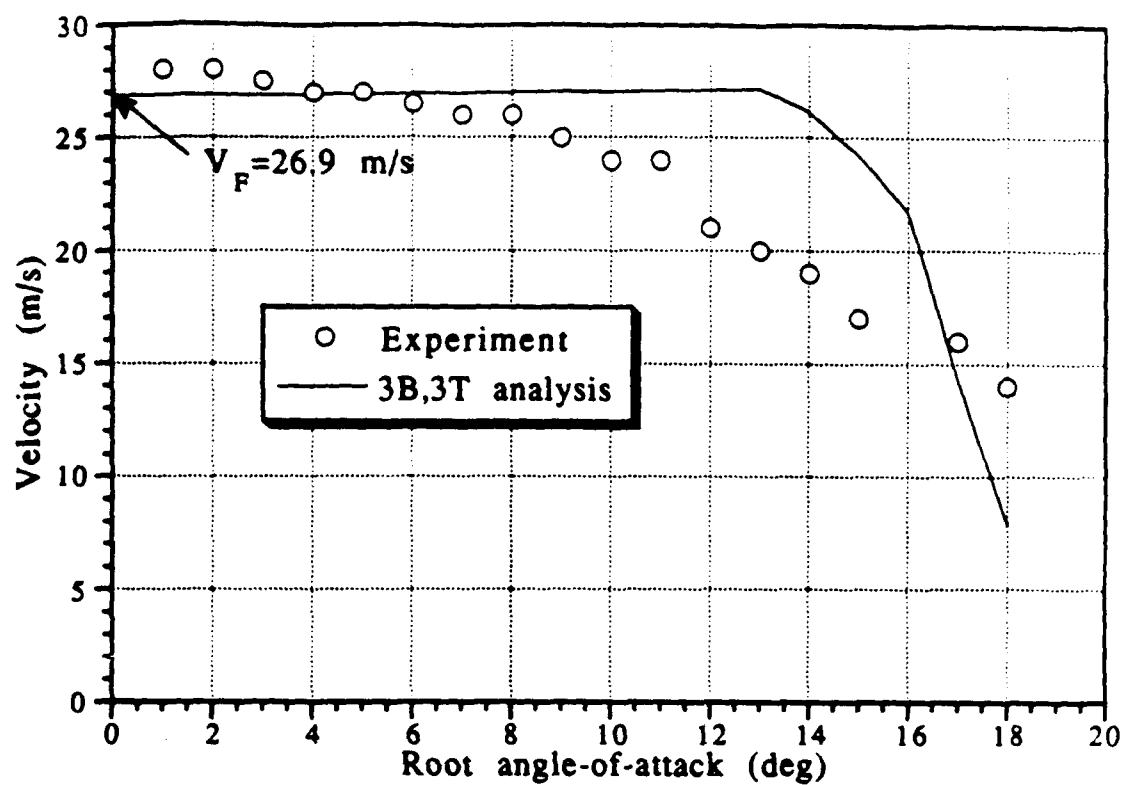
Fig. 25 $[+15_2/0_2]_S$ flutter boundary and frequency variation

135
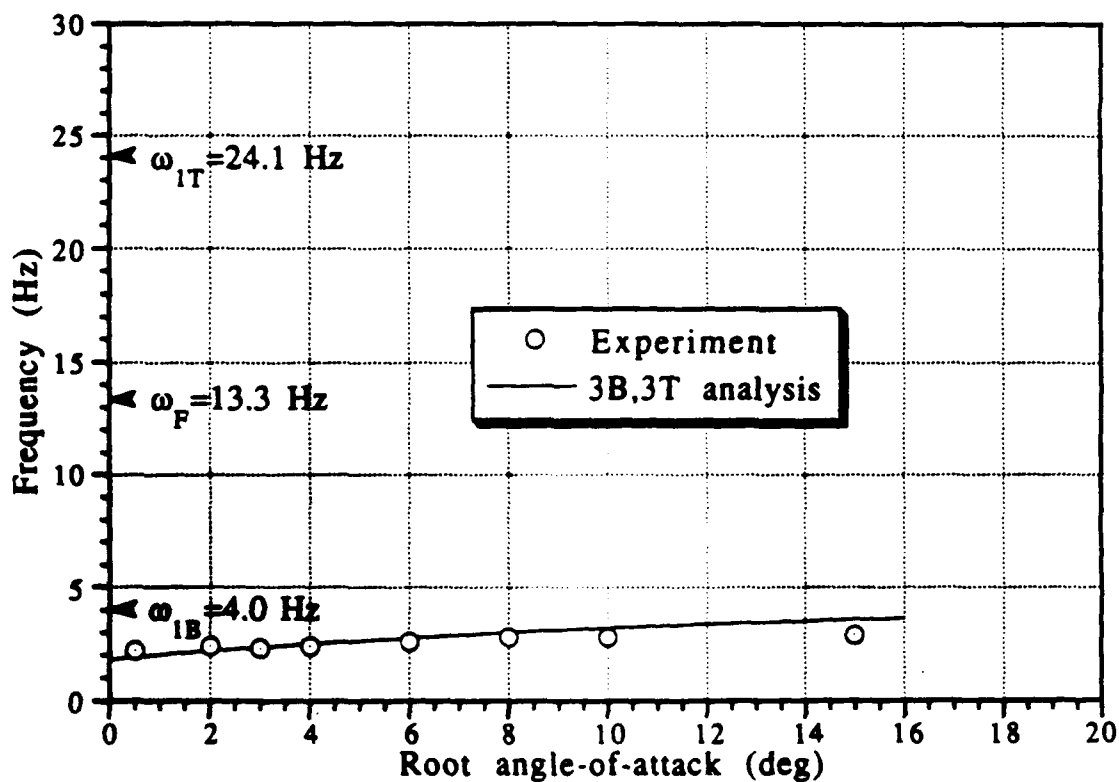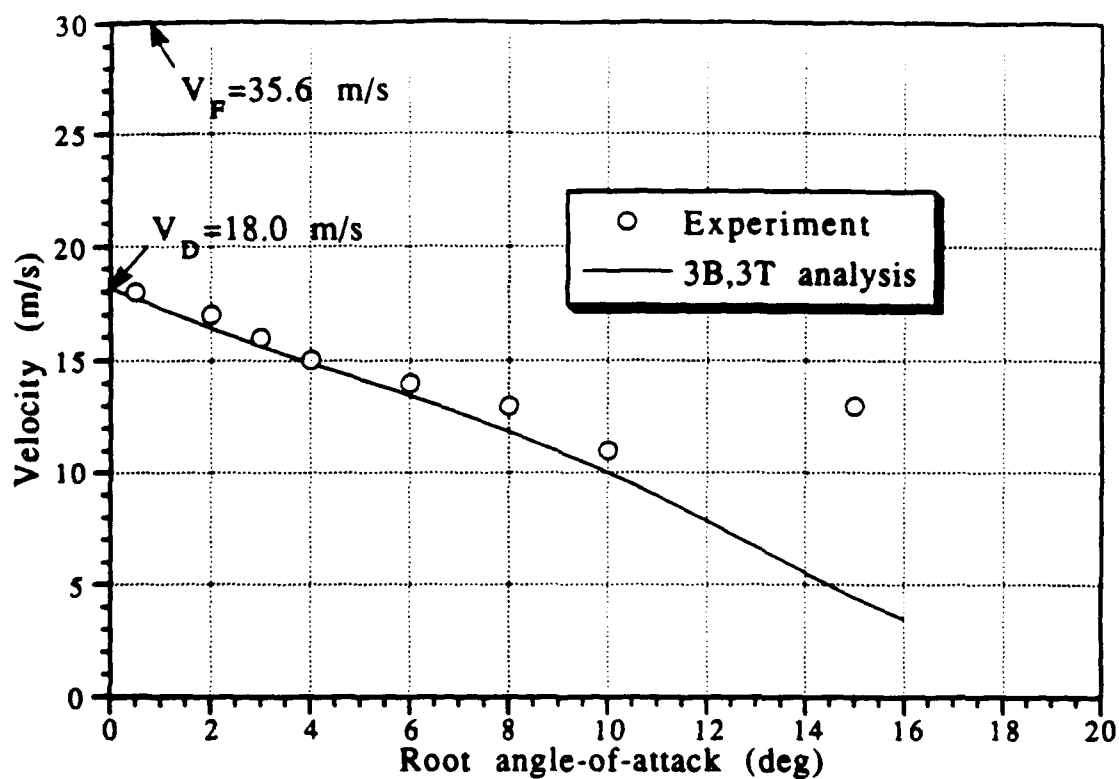
Fig. 26 $[-15_2/0_2]_S$ flutter boundary and frequency variation

5.2.3 Large-Amplitude, Nonlinear Flutter

The experimental and analytic flutter characteristics for increasing amplitudes of oscillation are presented in Figs. 27 to 32. The pairs of figures (Figs. 27 and 28 for the $[0_3/90]_S$ layup; Figs. 29 and 30 for the $[+15_2/0_2]_S$ layup; and Figs. 31 and 32 for the $[-15_2/0_2]_S$ layup) contain (a) the graphs of the time-averaged midchord tip deflection and of the time-averaged total tip angle (the sum of the root angle of attack and the tip twist), and (b) the graphs of the midchord tip deflection amplitude of oscillation and of the tip twist amplitude of oscillation, all for increasing velocity with constant root angle of attack ($\alpha_R = 1^\circ$, 5°, 10°, and 15°). For each line of constant root angle of attack α_R, both the full, unsteady flutter analysis (dashed lines on Figs. 27, 29, & 31; solid lines on Figs. 28, 30, & 32) and the steady, static analysis (unsteady terms suppressed) (solid lines on Figs. 27, 29, & 31) are presented, so as to show where the two meet (equivalent to the flutter boundary). Likewise, both the steady, static experimental data and the unsteady, flutter experimental data are presented. This experimental data is gathered from two sources: video recordings of tip deflections (hollow symbols) and converted strain gauge readings (solid symbols). (The root strain gauge bending and torsion readings were converted to approximate tip deflections by assuming the same shape for the spanwise distributed load as described in Appendix C).

Note that the analysis has been conducted without including fore-&-aft modes, only bending and torsion modes. This is because, as noted in previous sections, these modes have little effect on the

137

frequency and mode shape of the first torsional mode, which is the the dominant torsional mode in both linear and stalled flutter of the $[0_3/90]_S$ and $[+15_2/0_2]_S$ layups, and has no effect on the first bending mode, which is the dominant mode for the $[-15_2/0_2]_S$ layup.

Graphs in Fig. 27 for the $[0_3/90]_S$ laminate show the same trends in analysis: both the midchord tip deflection and total tip angle show a sharp decrease when the velocity is increased past the flutter boundary. These analytic trends compare favorably for root angles of attack $\alpha_R = 1^\circ$ & 5°, but less well for $\alpha_R = 10^\circ$ & 15°.

The averaged tip deflections for the $[0_3/90]_S$ layup show the typical characteristics of divergence. For low root angles of attack, the tip deflection remains relatively small just up until the point where the divergence speed is reached. If in fact the root angle of attack were $\alpha_R = 0^\circ$, then the deflection would remain zero until the divergence speed, then would "jump" to an asymptotic stalled behavior [Ref. 94]. For higher root angles of attack above the static stall angle, eg. $\alpha_R = 10^\circ$ & 15°, the tip deflection variations are almost identical — since the entire wing is past stall, the distributed load remains the same irrespective of the root angle of attack, and consequently the bending deflection is identical and only dependent on the increasing dynamic pressure.

The experimental values show identical trends as the analysis, although shifted slightly. At worst (i.e. at divergence/flutter for $\alpha_R = 1^\circ$ or 5°) they are 5-7 cm lower than the analytic predictions of 20-25 cm; near moderate deflections of approximately 10 cm the experimental values are 1-2 cm below the analysis. Since the behavior is highly nonlinear and involves rapid changes in deflec-

tions, these discrepancies between experiment and analysis seem within reason. However, the discrepancy between experimental values for $\alpha_R = 10°$ and $\alpha_R = 15°$ indicate that the assumption of unchanging static aerodynamic characteristics past stall is slightly inaccurate — since the experimental deflections for the higher root angle of attack are in fact larger, it can be deduced that the aerodynamic coefficients do not in fact level off, but continue to rise slightly. The trends in averaged total tip angle for the $[0_3/90]_S$ layup are essentially the same as for the averaged tip deflection, since the two are in fact coupled in producing the divergence phenomenon, but the correlation between analysis and experiment happens to be much closer.

The analysis in Fig. 27 predicts a sharp drop in both tip deflection and tip angle for $\alpha_R = 1°$ & $5°$ as the amplitude of oscillation increases past the flutter boundary. The drop in deflections is likely due to the dependence of mean load on amplitude of oscillation at low reduced frequency, as described in Section 5.1.4 for the 2D aerodynamic analysis. The reason the drop is so sudden is because, as noted in Section 5.2.2 for the flutter boundaries, when the root angle of attack is below the static stall angle the behavior is still essentially governed by the linear aerodynamics, so the oscillation growth is still essentially exponential up until deep stall is reached. Therefore, it would be expected that a small change in velocity would produce a large increase in amplitude of oscillation. This prediction matches the experiment fairly well for $\alpha_R = 1°$ & $5°$ in both Figs. 27 and 28. As the oscillation growth then continues within the deep stall region, the intuitive physical sense is that a hardening trend will

139

occur — that is, larger and larger velocities will be required to create larger amplitudes of oscillation. This assumption is borne out by the experiment, especially clear in the tip angle oscillation graph in Fig. 28, but is not well predicted by the analysis. Instead, the analysis shows hardening with a softening trend, that is, the increase in velocity required to produce the same increase in amplitude of oscillation progressively gets smaller and smaller. This is likely due to the poor prediction of the out-of-phase terms of the nonlinear aerodynamics at higher amplitudes of oscillation, as discussed in Section 5.1.4. The oscillation amplitude in bending in Fig. 28 indicates, as expected, that the nonlinear flutter phenomenon is single degree of freedom in torsion.

The characteristics of oscillation for the $[0_3/90]_S$ layup for $\alpha_R = 10°$ and $\alpha_R = 15°$ are essentially the same as for the low root angles of attack of $\alpha_R = 1°$ and $\alpha_R = 5°$, except that there is no sudden change through light stall to deep stall because the wings are already in deep stall from the root angle of attack. This means that the change in mean deflections is less pronounced than for $\alpha_R = 1°$ & $5°$, but still follows the trend of decreasing with increased amplitude of oscillation, again because of the same nonlinear aerodynamic effects. For example, at $\alpha_R = 1°$ an increase in velocity from 27 m/s to 28 m/s produces a 9° drop in twist, while at $\alpha_R = 10°$ an increase in velocity from 17 m/s to 22 m/s is required to produce the same drop. This tends not to match the tip deflection experimental data, which continues to increase, while the tip angle experimental data shows different trends for the video and strain gauge data.

140

The discrepancy in tip deflection indicates that there should be an increase in aerodynamic loading with oscillation amplitude in deep stall, while the trend might be the opposite in light stall. This discrepancy indicates a deficiency in the aerodynamic model that might be attributed to the "smearing" of the fixed-time stall delay. That is, it is convenient to smear the fixed-time stall delay over the hysteresis loop when the oscillation straddles the stall angle (i.e. light stalling), but it is probably inappropriate to do so when the entire hysteresis is above the stall angle and there is in fact *no fixed-time stall delay* (i.e. deep stall). The discrepancy in experimental data between video data and strain gauge data indicates that the assumption in converting the strain gauge data to tip angle is probably no longer valid — there are likely higher modes that are starting to come into play for $\alpha_R = 10^\circ$ & 15° that drastically alter the root curvatures. A better method of comparison would to compare the experimental strain readings against the predicted values derived from the root curvatures of the mode shapes.

Again, the experimental trends for $\alpha_R = 10^\circ$ & 15° for the $[0_3/90]_S$ layup follow physical intuition: they are already in deep stall so a hardening effect is observed such that greater and greater velocity is required to produce larger amplitudes of oscillation. Again, as with the low root angles of attack, the analysis predicts the initial growth fairly well but breaks down as the amplitude of oscillation grows from moderate to large, showing an analytic softening trend counter to the experimental hardening trend. And again, as with the low root angles of attack, the small bending oscillation

amplitudes indicate a single degree of freedom phenomenon in torsion.

Fig. 29 for the $[+15_2/0_2]_S$ layup shows the characteristics typical of negative bending-torsion coupling. Increasing velocity induces increased negative twist, as exemplified in the tip angle plot, thus precluding any sharp rise in tip deflection, as seen in the tip deflection plot. Even up to a root angle of attack of $\alpha_R = 15^\circ$, the flutter initiation is essentially governed by linear aerodynamics. Therefore, the oscillation growth is nearly exponential, until it is checked by the stalled aerodynamics.

This exponential growth behavior is most clear on the tip angle oscillation amplitude graph of Fig. 30, in which the oscillation amplitudes continue to increase at the same velocity until deep stall has been reached. The corresponding points on the averaged deflection plots of Fig. 29 are harder to identify — remembering that the velocity remains unchanged during the exponential growth, it then becomes clear that the tip deflections are also remaining unchanged, even as the oscillation amplitudes pass through light stall on the way to limit cycles at deep stall.

Once deep stall is reached, an analytic softening trend in the flutter characteristics is predicted — that is, once past the flutter boundary, the analysis predicts that a *decrease* in velocity will induce a jump to a larger flutter amplitude. This is a characteristic also observed in Refs. 94 and 95 for analysis with constant coefficients and is likely unfounded (and could not be reproduced experimentally), but might again be due to the breakdown of the 2-dimensional aerodynamic analysis at large amplitudes of oscilla-

tion. Also, as seen in Fig. 29, the flutter velocities for the $[+15_2/0_2]_S$ layup are very high, thus inducing large tip deflections. It is therefore likely that at high amplitudes of oscillation the governing effects are the nonlinear structural effects, not the nonlinear aerodynamic effects, and hardening effects will be observed such as those for cubic stiffening in Refs. 94 and 95.

The experimental trends for the $[+15_2/0_2]_S$ layup for increasing amplitudes of oscillation again show hardening effects, as would be expected from physical intuition. However, as seen in the graph for amplitude of oscillation in torsion, these hardening effects start to become noticeable even before the oscillation amplitudes reach the static stall angle. This might also be a consequence of the inaccurate static aerodynamic modeling below the stall angle, where in fact there is something like a parabolic drop before stall instead of a continuous linear rise. This would mean that the stalling effects would start to play a role in limiting the exponential growth even before the static stall angle were exceeded.

Fig. 31 for the $[-15_2/0_2]_S$ layup shows a more gentle deviation from the steady analysis, as compared to the sharp change in character demonstrated by the $[0_3/90]_S$ layup in Fig. 27. The experiment and analysis indicate that, contrary to the $[0_3/90]_S$ laminate, both the tip deflection and tip angle continue to increase once the velocity is increased past the flutter boundary. Essentially, however, the flutter characteristics are not unlike those of the $[0_3/90]_S$ layup. As with the $[0_3/90]_S$ layup, the steady characteristics of the $[-15_2/0_2]_S$ layup are principally governed by divergence phenomenon. The flutter at low root angles of attack is triggered by the transition into light stall.

143

Unlike the $[0_3/90]_S$ layup, however, it is difficult to compare how well the analysis compares against experiment for the averaged deflections past flutter initiation, since it is difficult to tell how much the decrease in load with increasing amplitude of oscillation is counteracted by the increase in deflection from increasing velocity. All that can be concluded from Fig. 31 is that the deflections generally continue to increase past the flutter boundary, both analytically and experimentally, thus indicating that for the $[-15_2/0_2]_S$ layup the divergence effects tend to overpower the nonlinear coupling between oscillation and mean aerodynamic load.

Fig. 32 for the $[-15_2/0_2]_S$, showing the amplitude of oscillation of the bending and torsion components, displays a much higher bending component, as would be expected for this predominantly bending stall flutter. In contrast to the $[0_3/90]_S$ and $[+15_2/0_2]_S$ layups, the analysis shows more hardening than the experimental trends — that is, the bending oscillation amplitudes grow very quickly from just a small increase in velocity, while the analysis would predict that a larger increase in velocity would be required. This discrepancy might be due to two possible effects. First, the dependence of the nonlinear aerodynamics on the difference between perceived angle of attack due to pitch (θ) and due to plunge rate ($\dot{h}/U$) might not be accurately modeled, despite the physical arguments presented in previous investigations [Refs. 55 & 56]. Second, while the geometric nonlinearities have been taken into account for large mean deflections, the current analysis does not take into account large amplitudes of oscillation of the deflections, only small amplitudes of oscillation around a large deflected mean. These

144

two effects might combine so as to account for the difference in softening between the experimental and analytic values. Again, as with the previous layups, large amplitudes of oscillation tend to produce a lessening of the hardening in the analysis (i.e. the hardening with velocity is smaller at large amplitudes of oscillation than at small amplitudes of oscillation) that is not at all evident experimentally. Also, as noted for the flutter boundary analysis, the values at root angle of attack $\alpha_R = 15^\circ$ are likely so disparate because of large experimental errors.
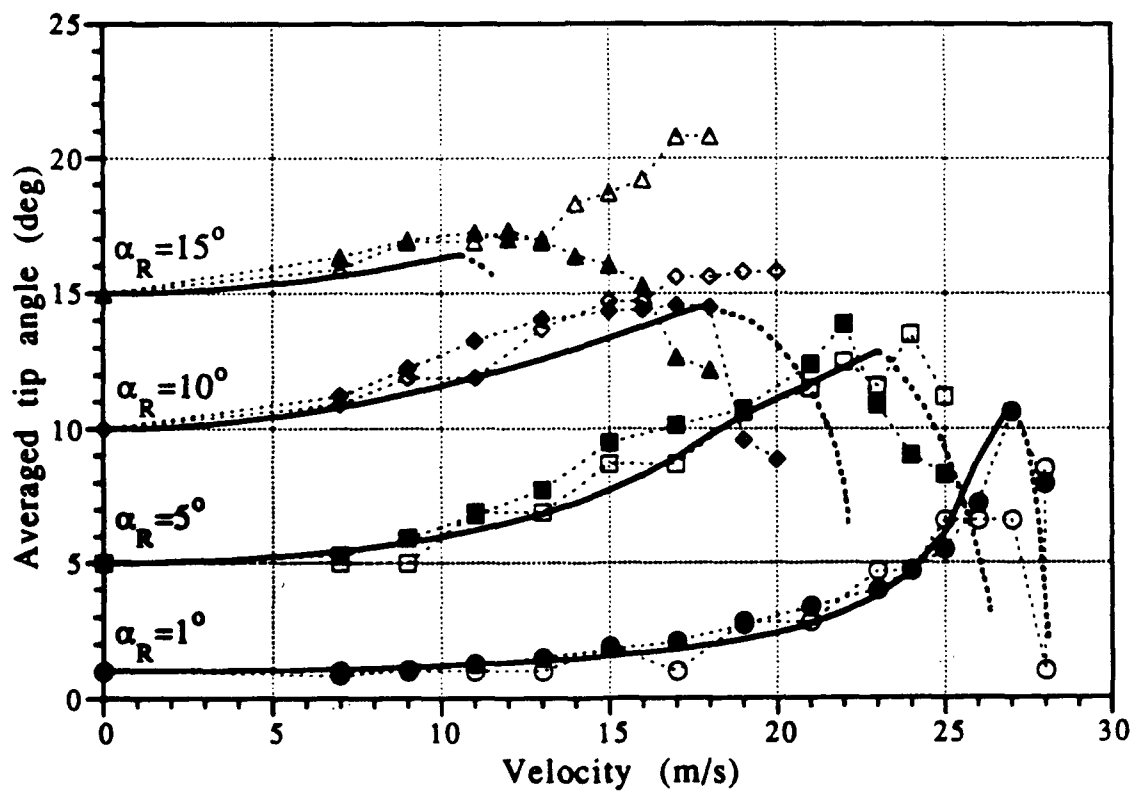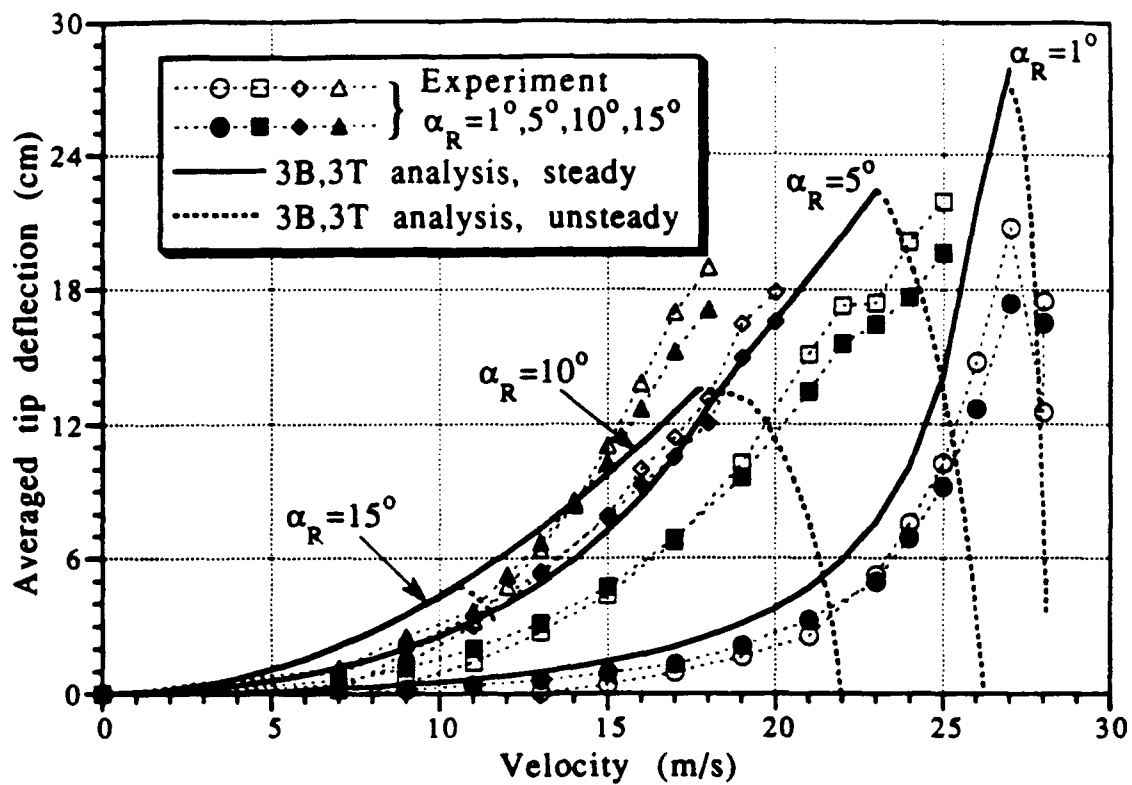
Fig. 27 $[0_3/90]_S$ averaged midchord tip deflections, and averaged total tip angle

146
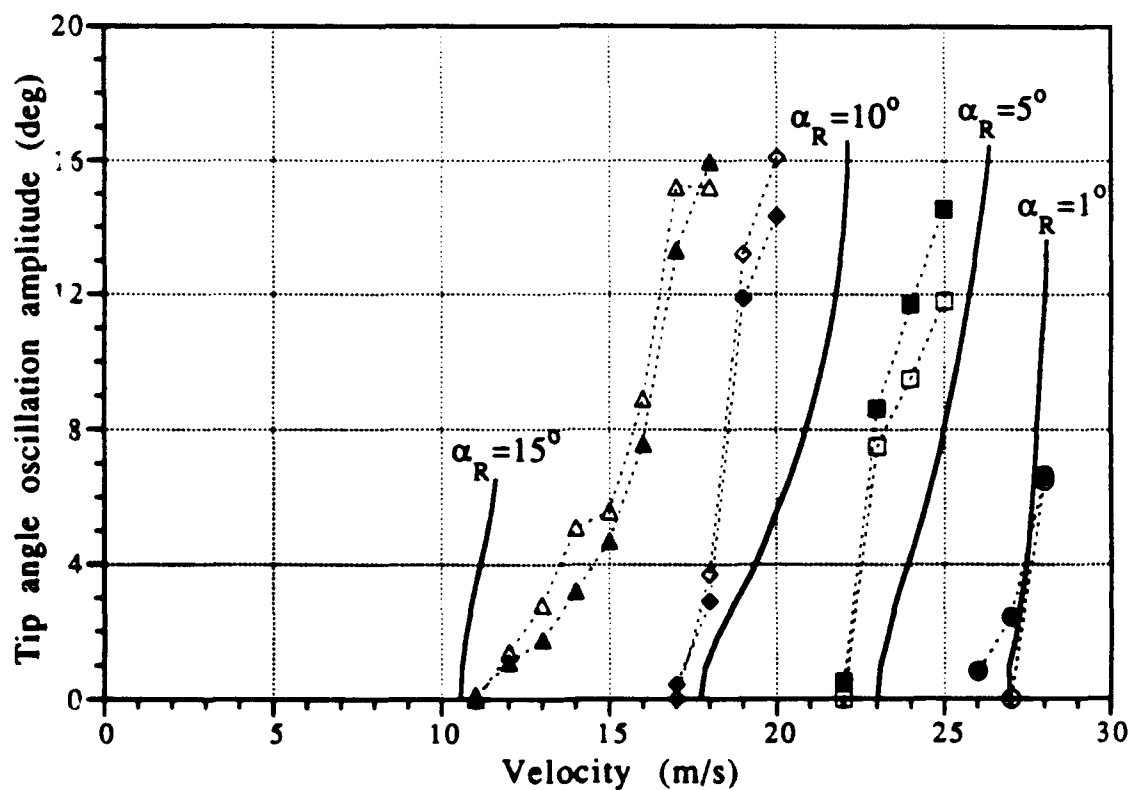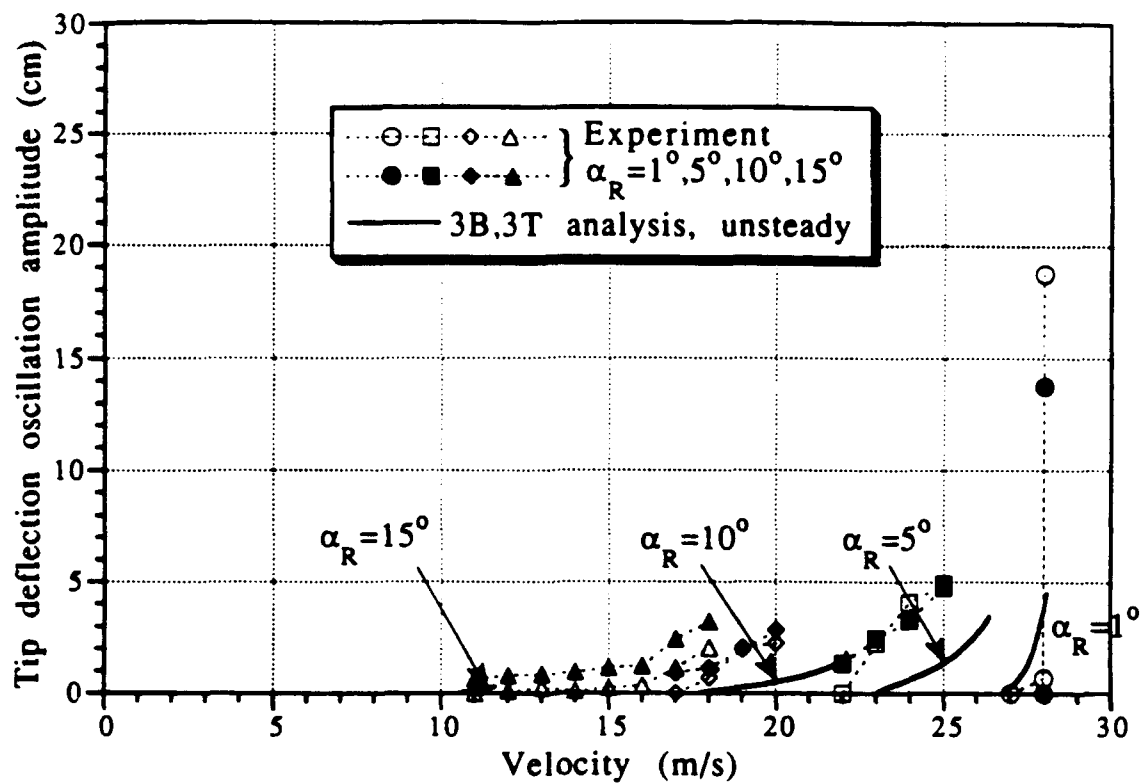
Fig. 28 $[0_3/90]_S$ tip deflection and tip angle oscillation amplitudes
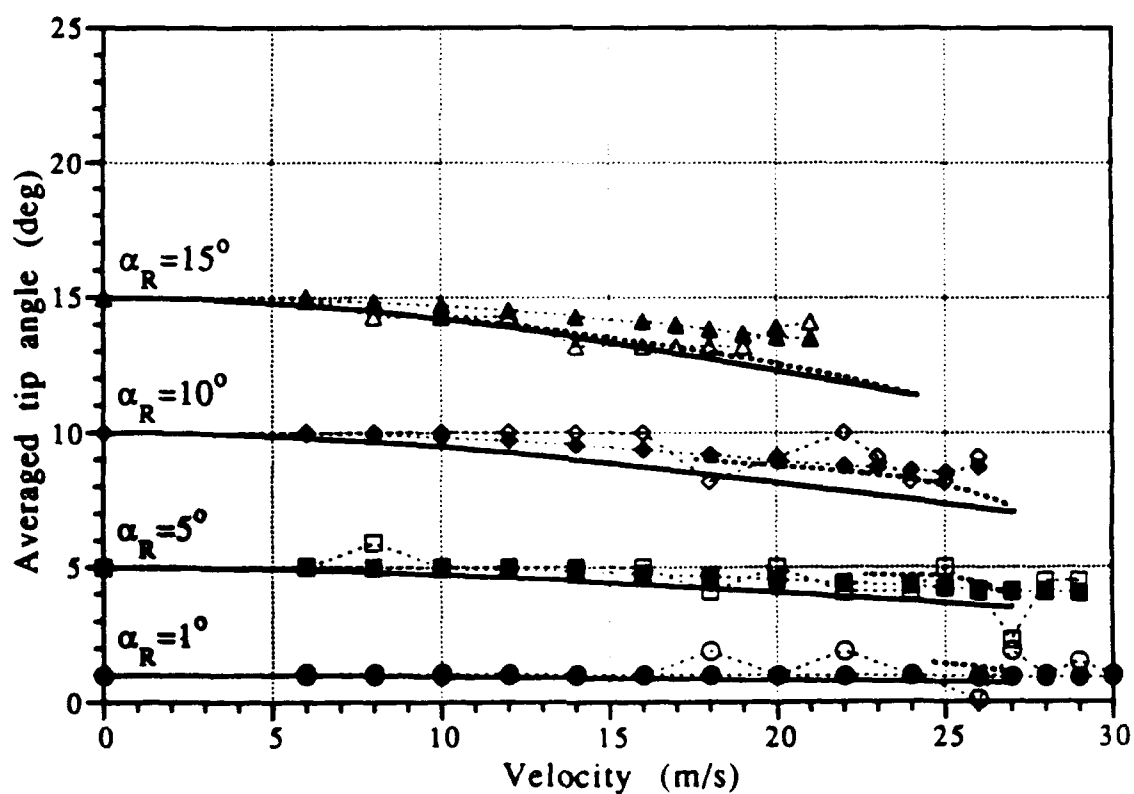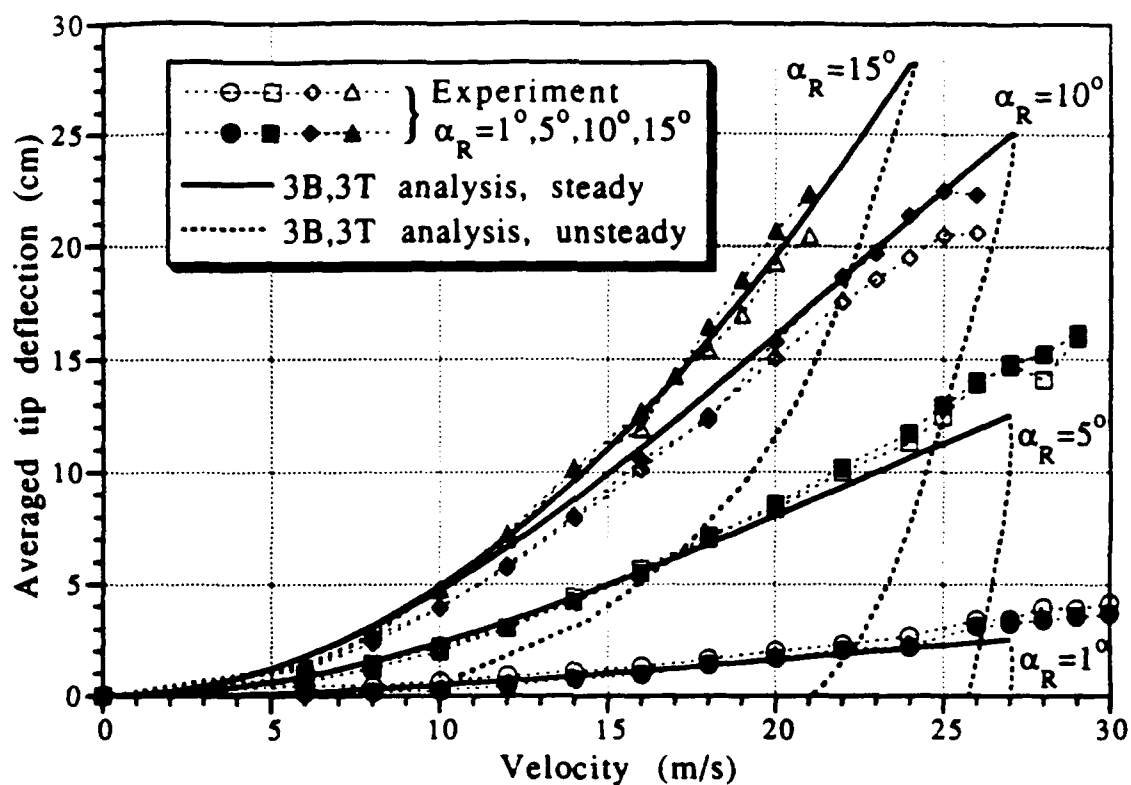
Fig. 29 $[+15_2/0_2]_S$ averaged midchord tip deflections, and averaged total tip angle

148
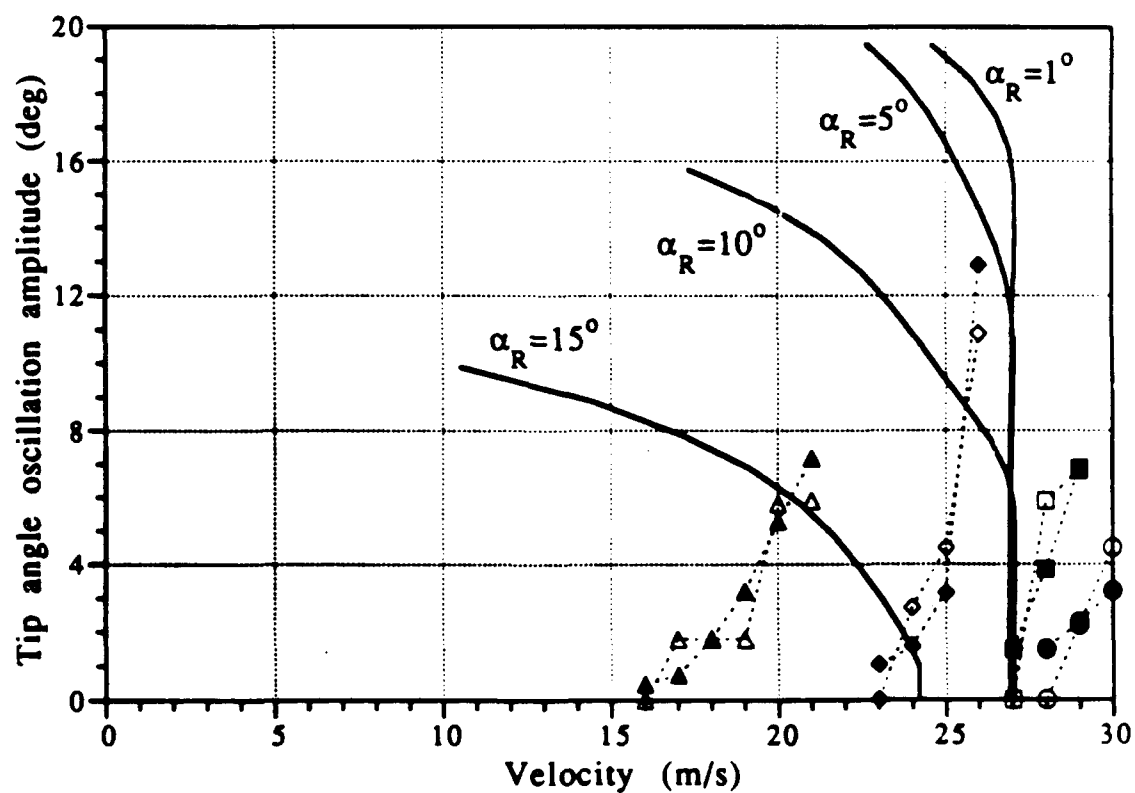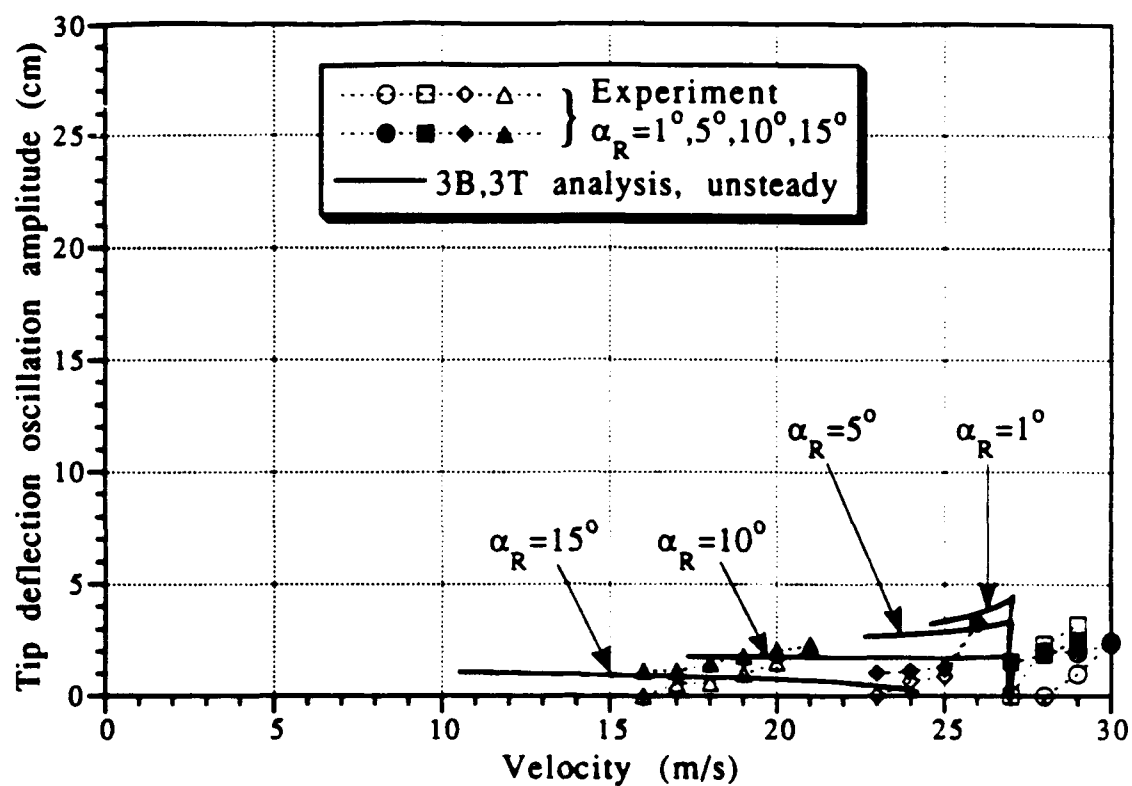
Fig. 30 $[+15_2/0_2]_S$ tip deflection and tip angle oscillation amplitudes
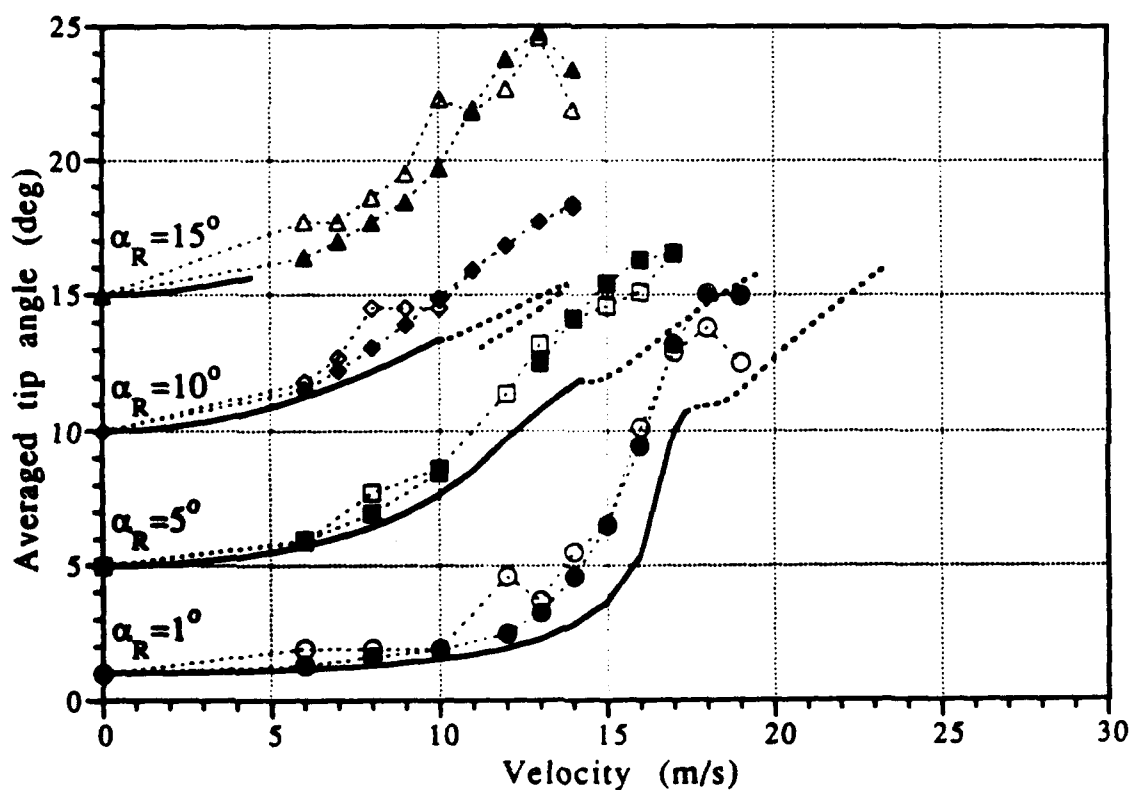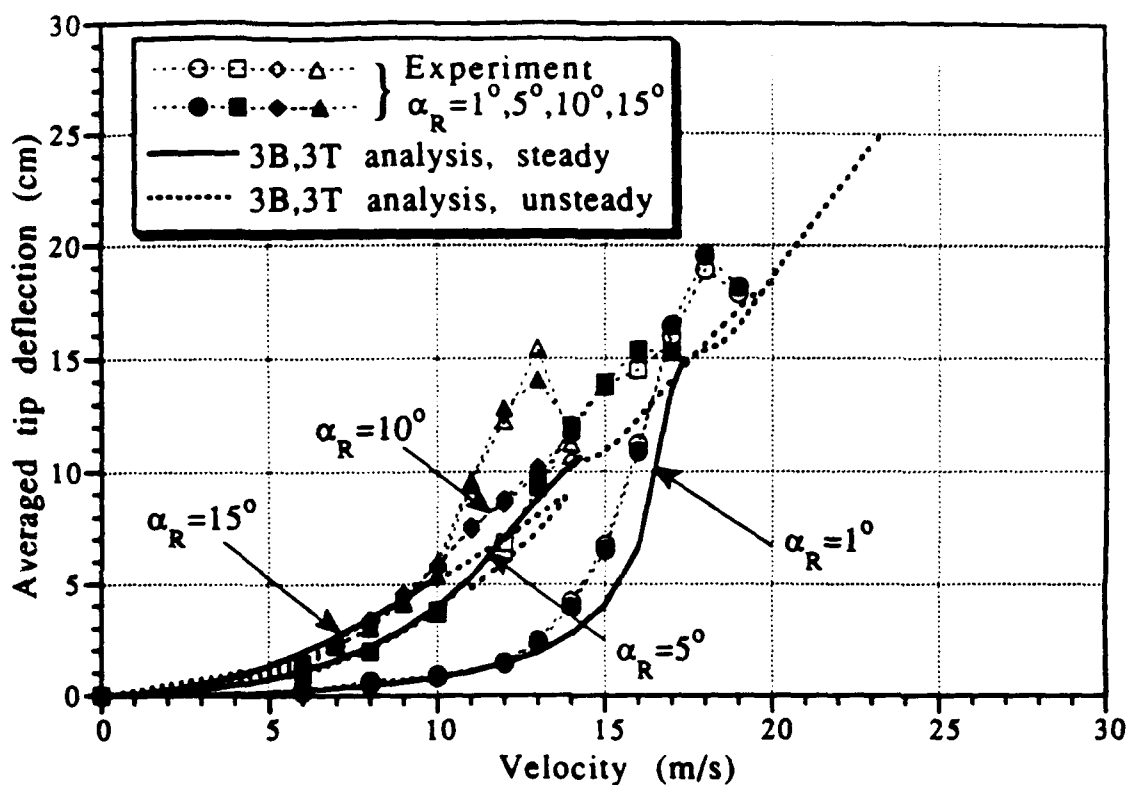
Fig. 31 $[-15_2/0_2]_S$ averaged midchord tip deflections, and averaged total tip angle

150
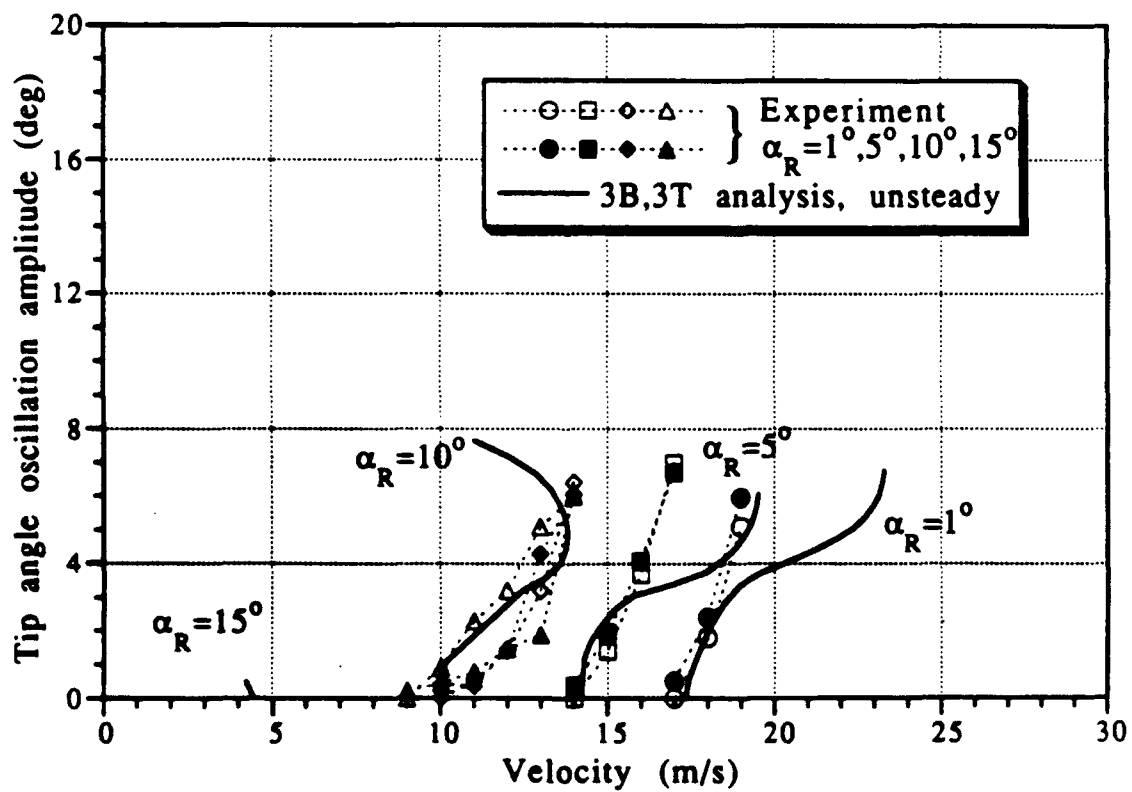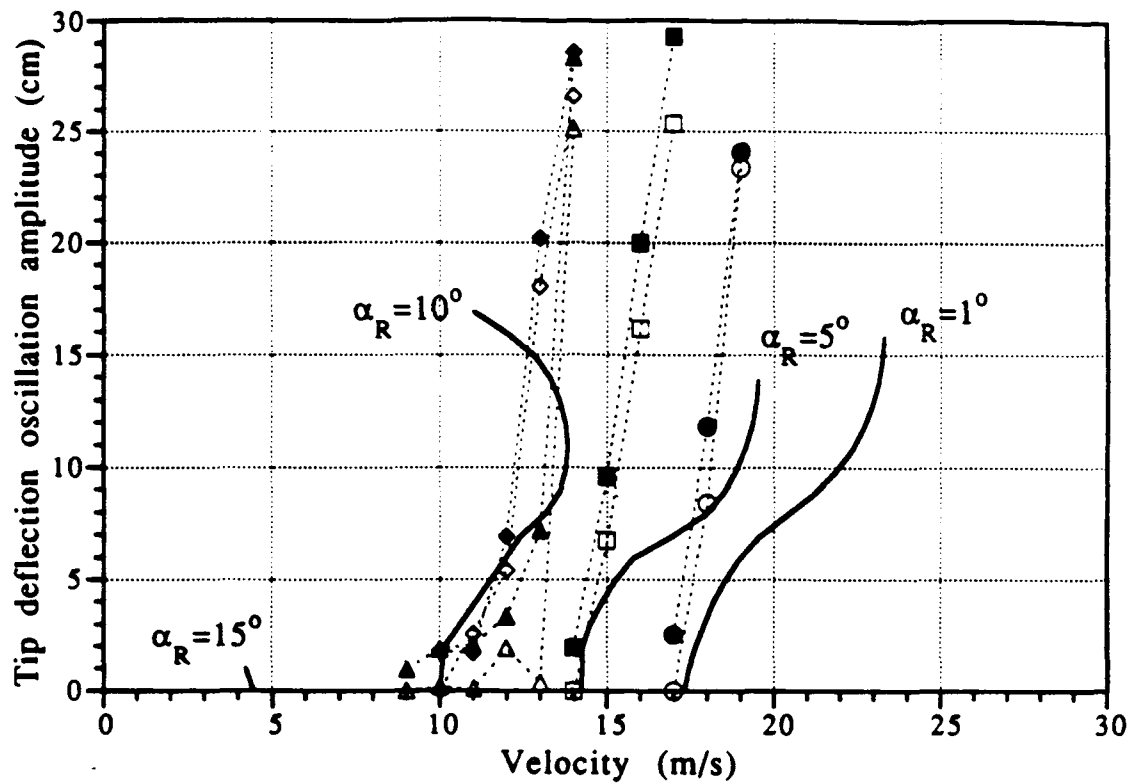
Fig. 32 $[-15_2/0_2]_S$ tip deflection and tip angle oscillation amplitudes

Chapter VI
Conclusions & Recommendations

An analytic method has been developed to include nonlinear structural and nonlinear aerodynamic effects into a full, 3-dimensional, aeroelastic problem, using the mathematical tools of Fourier analysis, harmonic balance, and the Newton-Raphson method as a numerical solver. The method makes use of the geometrically nonlinear, Hodges & Dowell structural model, based on a second-order ordering scheme, together with the ONERA stall flutter model for the aerodynamics. Although in the current investigation the method is used with many simplifications — for example in the simplification of the aerodynamic force curves, in the semi-empirical nature of the aerodynamic model, and in the low number of harmonics used in the harmonic balance method — the formulation can be extended to implement more complex variations of these factors. The current analysis extends on previous work by more thoroughly investigating the effects of nonlinear, large amplitude deflections, and by more accurately modeling the nonlinear aerodynamics of the ONERA model within the context of a harmonic balance scheme.

As shown in Chapter 5, the current nonlinear aeroelastic analysis predicts well almost all the observed, experimental, nonlinear stall phenomena. Specifically, flutter boundaries have been obtained which decrease with root angle of attack, limit cycle amplitudes at flutter have been obtained, and the transitions from linear, bending-torsion flutter to torsional stall flutter, and from linear divergence to bending stall flutter, have been predicted analytically. In addition,

153

within the range of the valid amplitudes of oscillation for the ONERA model, the analysis correctly predicts the experimental hardening trend as amplitude of oscillation increases.

6.1 Aerodynamic Model

The current investigation has contributed a unique approach to the application of the ONERA model to stall flutter analyses and has many advantages. First and foremost, by the use of the ONERA model, the method is in such a form that it is generalizable for a wide range of parameters — such as airfoil type and Reynolds number, as long as the aerodynamic characteristics of the airfoil are available — and thus relieves some of the cumbersomeness inherent in purely theoretical models. Second, by the application of harmonic balance and Rayleigh-Ritz to the ONERA model, the method is in a simplified form that allows the user to choose the number of mode shapes or order of harmonics to suit his particular problem, while retaining the full nonlinearity of the formulation. Third, by use of Fourier analysis and harmonic balance, the current analysis avoids the need for time-marching integration and avoids any computational time that might be needed in such a method to reach the final flutter limit cycle.

However, as currently implemented, the model still has limitations (other than the limitations already inherent in a semi-empirical model such as the ONERA model). First, the current application of Fourier analysis to the forcing terms ignores the fixed-time stall delay of the ONERA model. Second, as noted in Section 5.1.4, the current model accurately reproduces the exact time marching solution to the ONERA equations for moderate amplitudes of oscillation, but

breaks down for larger amplitudes. This precludes properly predicting flutter characteristics well beyond the small amplitude flutter boundary. Third, there is little low Reynolds number data from which to extract the ONERA nonlinear coefficients.

Fortunately, these deficiencies are not inherent to the model itself, but are reflections of its current mode of application. The fixed-time stall delay can be directly implemented — instead of being "smeared" over the entire hysteresis loop — by incorporating a Fourier series step function multiplied by the current formulation, so as to turn "off" the nonlinearity during the appropriate lag time. Larger amplitudes of oscillation can be handled by applying the harmonic balance with a larger number of harmonics. However, as discussed in Section 3.4.3, using a time marching analysis would probably be more computationally efficient if more than two harmonics are required. More accurate coefficients can be determined by simply running the appropriate 2-dimensional aerodynamic tests with the current wing specimens (so as to retain the correct surface roughness, et cetera), although this is a recommendation that is applicable to the current investigation only, and would not be necessary for applications to real, operational devices, for which much data already exists at the appropriate Reynolds number. Further work also needs to be done in determining 3-dimensional aerodynamic effects, although little work in semi-empirical models has yet been accomplished in this domain.

6.2 Structural Model

The current investigation has added two contributions to the theory of ordering schemes for application to nonlinear structural modeling. First, it has extended the application of the Hodges & Dowell nonlinear equations to the realm of anisotropic materials and, more generally, has outlined the scheme by which those equations can be implemented for beams with through-the-thickness variation or through-the-thickness asymmetry. Second, it has shown by comparison to Minguet's Euler angle/Finite Difference method that a modal approach to the Hodges & Dowell nonlinear equations yields satisfactory results, provided that sufficient fore-&-aft modes are used, making those equations more tractable in aeroelastic applications.

Unfortunately, it was found in the current investigation that *the contribution of the nonlinear structures to this particular aero-elastic problem was insignificant.* It would be interesting to make a further analytic and experimental investigation with wings that were less stiff in the fore-&-aft direction, i.e. which had fore-&-aft frequencies much closer to the bending and torsion frequencies. Such an investigation might be accomplished with wings that were more square in cross section, instead of low thickness-to-chord ratio.

From a theoretical viewpoint, it still remains to somehow analytically model the cubic stiffening observed in previous investigations, since this effect seems to play a large role in the hardening phenomenon observed at larger amplitudes of oscillation. It might also be interesting to further delve into the effects of chordwise asymmetry (while the through-the-thickness asymmetry has already

156

been covered), since this is taken into account by the Hodges & Dowell equations but ignored in the current analysis because of the low stiffness of the asymmetrical NACA 0012 styrofoam fairings.

6.3 Experiment

Experimental data have been obtained on a set of aeroelastically tailored wings with varying amounts of bending-torsion coupling and matched the trends of previous studies [Refs. 90 to 92]. A more in-depth experimental investigation of the transition from linear to nonlinear flutter behavior has been accomplished, and a more extended set of data past the flutter boundary has been collected.

As mentioned in Section 6.1, it would be desirable to experimentally investigate the 2-dimensional aerodynamic behavior of these same wings, so as to fine tune the ONERA model. Also, as mentioned in Section 6.2, it might also be desirable to make a more thorough investigation of the structural nonlinearities by running experiments with wings that are softer in the fore-&-aft direction. Beyond these recommendations, further work might be focused toward the investigation of the variation of other parameters affecting stall flutter with composite wings: taper or spanwise variation of other properties (such as layup or stiffness); fore and aft sweep, with large deflection; the aerodynamic and structural effects of stores and fuselage; the nonlinear aerodynamic and structural effects within body freedom flutter.

157

References

Dynamic Stall Experiments:

1. Liiva, J., and Davenport, F.J., "Dynamic Stall of Airfoil Sections for High-Speed Rotors," *Journal of the American Helicopter Society*, Vol. 14, No. 2, April 1969, pp. 26-33.

2. McAlister, K.W., Carr, L.W., and McCroskey, W.J., "Dynamic Stall Experiments on the NACA 0012 Airfoil," NASA TP-1100, January 1978.

3. McCroskey, W.J., McAlister, K.W., Carr, L.W., Pucci, S.L., Lambert, O., and Indergrand, R.F., "Dynamic Stall on Advanced Airfoil Sections," *Journal of the American Helicopter Society*, Vol. 26, No. 3, July 1981, pp. 40-50.

4. McAlister, K.W., Pucci, S.L., McCroskey, W.J., and Carr, L.W., "An Experimental Study of Dynamic Stall on Advanced Airfoil Sections Volume 1: Summary of the Experiment," NASA TM-84245, July 1982.

5. McAlister, K.W., Pucci, S.L., McCroskey, W.J., and Carr, L.W., "An Experimental Study of Dynamic Stall on Advanced Airfoil Sections Volume 2: Pressure and Force Data," NASA TM-84245, September 1982.

Dynamic Stall Flow Visualization:

6. Carta, F.O., "Analysis of Oscillatory Pressure Data Including Dynamic Stall Effects," NASA CR-2394, 1974.

7. McAlister, K.W., and Carr, L.W., "Water-Tunnel Experiments on an Oscillating Airfoil at Re=21,000," NASA TM-78446, March 1978.

8. McAlister, K.W., and Carr, L.W., "Water-Tunnel Visualizations of Dynamic Stall," *Trans. ASME, Journal of Fluids Engineering*, Vol. 101, 1979, pp. 376-380.

Identification of Processes in Dynamic Stall:

9. Carr, L.W., McAlister, K.W., and McCroskey, W.J., "Analysis of the Development of Dynamic Stall Based on Oscillating Airfoil Experiments," NASA TN-D-8382, January 1977. (Also, McCroskey, W.J., Carr, L.W., and McAlister, K.W., "Dynamic Stall Experiments on Oscillating Airfoils," *AIAA Journal*, Vol. 14, No. 1, January 1976, pp. 57-63.)

10. McCroskey, W.J., "The Phenomenon of Dynamic Stall," NASA TM-81264, March 1981.

Discrete Potential Vortex Methods for Dynamic Stall:

11. Ham, N.D., and Garelick, M.S., "Dynamic Stall Considerations in Helicopter Rotors," *Journal of the American Helicopter Society*, Vol. 13, No. 2, April 1968, pp. 49-55.

12. Ham, N.D., "Aerodynamic Loading on a 2-Dimensional Airfoil During Dynamic Stall," *AIAA Journal*, Vol. 6, No. 10, October 1968, pp. 1927-1934.

13. Baudu, N., Sagner, M., and Souquet, J., "Modélisation du Déchrochage Dynamique d'un Profil Oscillant," AAAF 10ième Colloque d'Aéronautique Apliquée, Lille, France, 1973.

14. Giesing, J.P., "Nonlinear 2-Dimensional Potential Flow with Lift," *Journal of Aircraft*, Vol. 5, No. 2, March/April 1968, pp. 135-143.

15. Ono, K., Kuwahara, K., and Oshima, K., "Numerical Analysis of Dynamic Stall Phenomena of an Oscillating Airfoil by the Discrete Vortex Approximation," Paper No. 8, 7th International Conference on Numerical Methods in Fluid Dynamics, Stanford, California, 1980.

16. Katz, J. "A Discrete Vortex Method for the Nonsteady Separated Flow Over an Airfoil," *Journal of Fluid Mechanics*, Vol. 102, January 1981, pp. 315-328.

17. Spalart, P.R., Leonard, A., and Baganoff, D., "Numerical Simulation of Separated Flows," NASA TM-84328, February 1983.

Zonal Methods for Dynamic Stall:

18. Crimi, P., and Reeves, B.L., "A Method for Analyzing Dynamic Stall of Helicopter Rotor Blades," NASA CR-2009, May 1972. (Also, AIAA-72-37, January 1972.)

19. Crimi, P., "Investigation of Nonlinear Inviscid and Viscous Flow Effects in the Analysis of Dynamic Stall," NASA CR-2335, February 1974.

20. Rao, B.M., Maskew, B., and Dvorak, F.A., "Theoretical Prediction of Dynamic Stall on Oscillating Airfoils," American Helicopter Society Paper 78-62, 1978.

21. Maskew, B., and Dvorak, F.A., "Investigation of Separation Models for the Prediction of $C_{L_{max}}$," *Journal of the American Helicopter Society*, Vol. 23, No. 2, April 1978, pp. 2-8.

22. Scruggs, R.M., Nash, J.R., and Singleton, R.E., "Analysis of Flow Reversal Delay for a Pitching Airfoil," AIAA-74-183, January 1974.

23. Chi, R.M., "Separated Flow Unsteady Aerodynamic Theory," *Journal of Aircraft*, Vol. 22, No. 11, November 1985, pp. 956-964.

Navier-Stokes Methods for Dynamic Stall:

24. Mehta, U.B., "Dynamic Stall of an Oscillating Airfoil," AGARD CP-277, AGARD, France, 1977, pp. 23-1 to 23-32. (Also, AGARD Fluid Dynamics Panel Symposium on Unsteady Aerodynamics, Paper No. 23, Ottawa, Canada, September 1977.)

25. Shamroth, S.J., and Gibeling, H.J., "The Prediction of the Turbulent Flow Field about an Isolated Airfoil," AIAA-79-1543, paper presented at the AIAA 12th Fluid & Plasma Dynamics Conference, Williamsburg, Virginia, July 1979.

26. Shamroth, S.J., and Gibeling, H.J., "A Compressible Solution of the Navier-Stokes Equations for Turbulent Flow About an Airfoil," NASA CR-3183, October 1979.

27. Sugavanum, A., and Wu, J.C., "Numerical Study of Separated Turbulent Flow Over Airfoils," AIAA-80-1441, paper presented at the AIAA 13th Fluid & Plasma Dynamics Conference, Snowmass, Colorado, July 1980.

28. Shamroth, S.J., "Calculation of Steady and Unsteady Airfoil Flow Fields via the Navier Stokes Equations," NASA CR-3899, August 1985.

29. Sankar, N.L., and Tang, W., "Numerical Solution of Unsteady Viscous Flow Past Rotor Sections," AIAA Paper 85-0129, January 1985.

30. Rumsey, C.L., and Anderson, W.K., "Some Numerical and Physical Aspects of Unsteady Navier-Stokes Computations Over Airfoils Using Dynamic Meshes," AIAA-88-0329, paper presented at the AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 1988.

Boeing-Vertol Method for Dynamic Stall:

31. Gross, D.W., and Harris, F.D., "Prediction of Inflight Stalled Airloads from Oscillating Airfoil Data," American Helicopter Society 25th Annual National Forum, American Helicopter Society, 1969.

32. Harris, F.D., Tarzanin, F.J., Jr., and Fisher, R.K., Jr., "Rotor High-Speed Performance; Theory vs. Test," *Journal of the American Helicopter Society*, Vol. 15, No. 3, July 1970, pp. 35-44.

33. Gormont, R.E., "A Mathematical Model of Unsteady Aerodynamics and Radial Flow for Application to Helicopter Rotors," US Army AMRDL TR-72-67, May 1973.

UTRC Method for Dynamic Stall:

34. Carta, F.O., et al., "Analytical Study of Helicopter Rotor Stall Flutter," American Helicopter Society 26th Annual National Forum, American Helicopter Society, 1970.

35. Carta, F.O., et al., "Investigation of Airfoil Dynamic Stall and its Influence on Helicopter Control Loads," US Army AMRDL TR-72-51, 1972.

36. Carta, F.O., and Carlson, R.G., "Determination of Airfoil and Rotor Blade, Dynamic Stall Response," *Journal of the American Helicopter Society*, Vol. 18, No. 2, April 1973, pp. 31-39.

Synthesization Methods for Dynamic Stall:

37. Bielawa, R.L., "Synthesized Unsteady Airfoil Data with Applications to Stall Flutter Calculations," American Helicopter Society 31st Annual National Forum, American Helicopter Society, May 1975.

38. Gangwani, S.T., "Prediction of Dynamic Stall and Unsteady Airloads for Rotor Blades," *Journal of the American Helicopter Society*, Vol. 27, No. 4, October 1982, pp. 57-64.

39. Gangwani, S.T., "Synthesized Airfoil Data Method for Prediction of Dynamic Stall and Unsteady Airloads," NASA CR-3672, February 1983.

MIT Method for Dynamic Stall:

40. Johnson, W., "The Effect of Dynamic Stall on the Response and Airloading of Helicopter Rotor Blades," *Journal of the American Helicopter Society*, Vol. 14, No. 2, April 1969, pp. 68-79.

41. Johnson, W., and Ham, N.D., "On the Mechanism of Dynamic Stall," *Journal of the American Helicopter Society*, Vol. 17, No. 4, October 1972, pp. 36-45.

Lockheed Method for Dynamic Stall:

42. Ericsson, L.E., and Reding, J.P., "Dynamic Stall of Helicopter Blades," *Journal of the American Helicopter Society*, Vol. 17, No. 1, January 1972, pp. 11-19.

43. Ericsson, L.E., and Reding, J.P., "Stall Flutter Analysis," *Journal of Aircraft*, Vol. 10, No. 1, January 1973, pp. 5-13.

44. Ericsson, L.E., and Reding, J.P., "Dynamic Stall Analysis in the Light of Recent Numerical and Experimental Results," *Journal of Aircraft*, Vol. 13, No. 4, April 1976, pp. 248-255.

45. Ericsson, L.E., and Reding, J.P., "Dynamic Stall at High Frequency and Large Amplitude," *Journal of Aircraft*, Vol. 17, No. 3, March 1980, pp. 136-142.

Time Delay Methods for Dynamic Stall:

46. Beddoes, T.S., "A Synthesis of Unsteady Aerodynamic Effects Including Stall Hysteresis," *Vertica*, Vol. 1, No. 2, 1976, pp. 113-123.

47. Beddoes, T.S., "Prediction Methods for Unsteady Separated Flows," Special Course in Unsteady Aerodynamics, AGARD R-679, France, March 1980, pp. 15-1 to 15-11..

48. Carlson, et al., "Dynamic Stall Modeling and Correlation with Experimental Data on Airfoils and Rotors," Paper No. 2, NASA SP-352, 1974.

ONERA Method for Dynamic Stall:

49. Tran, C.T., and Petot, D., "Semi-Empirical Model for the Dynamic Stall of Airfoils in View of Application to the Calculation of Responses of a Helicopter in Forward Flight," *Vertica*, Vol. 5, No. 1, 1981, pp. 35-53.

50. Dat, D., and Tran, C.T., "Investigation of the Stall Flutter of an Airfoil with a Semi-Empirical Model of 2-D Flow," *Vertica*, Vol. 7, No. 2, 1983, pp. 73-86.

51. Petot, D., and Loiseau, H., "Successive Smoothing Algorithm for Constructing the Semi-Empirical Model Developed at ONERA to Predict Unsteady Aerodynamic Forces," NASA TM-76681, March 1982.

52. Petot, D., "Dynamic Stall Modeling of the NACA 0012 Profile," Short Note, *Recherches Aérospatiales*, 1984-6, pp. 55-58.

53. McAlister, K.W., Lambert, O., and Petot, D., "Application of the ONERA Model of Dynamic Stall," NASA Technical Paper 2399, AVSCOM Technical Report 84-A-3, November 1984.

54. Petot, D., and Dat, R., "Unsteady Aerodynamic Loads on an Oscillating Airfoil with Unsteady Stall," 2nd Workshop on Dynamics and Aeroelasticity Stability Modeling of Rotorcraft Systems, Florida Atlantic University, Boca Raton, Florida, November 1987.

165

Pitch/Plunge Distinction for Dynamic Stall:

55. Peters, D.A., "Toward a Unified Lift Model for Use in Rotor Blade Stability Analyses," *Journal of the American Helicopter Society,* Vol. 30, No. 3, July 1985, pp. 32-42.

56. Rogers, J.P., "Applications of an Analytic Stall Model to Time-History and Eigenvalue Analysis of Rotor Blades," *Journal of the American Helicopter Society,* Vol. 29, No. 1, January 1984, pp. 25-33.

57. Fukushima, T., and Dadone, L.U., "Comparison of Dynamic Stall Phenomena for Pitching and Vertical Translation Motions," NASA CR-2693, 1977.

58. Carta, F.O., "A Comparison of the Pitching and Plunging Response of an Oscillating Airfoil," NASA CR-3172, October 1979.

59. Ericsson, L.E., and Reding, J.P., "The Difference Between the Effects of Pitch and Plunge on Dynamic Airfoil Stall," 9th European Rotorcraft Forum, Stresa, Italy, September 1983, pp. 8-1 to 8-8.

Summaries of Dynamic Stall Methods:

60. Johnson, W., "Comparison of Three Methods for Calculation of Helicopter Rotor Blade Loading and Stresses Due to Stall," NASA TN-D-7833, November 1974.

61. Philippe, J.J., "Dynamic Stall: An Example of Strong Interaction Between Viscous and Inviscid Flow," NASA TM-75447, 1978.

62. Johnson, W., *Helicopter Theory,* Chapter 16, Princeton University Press, 1980.

63. Carr, L.W., "Dynamic Stall Progress in Analysis and Prediction," AIAA Paper 85-1769, August 1985.

64. Galbraith, R.A.M., and Vezza, M., "Methods of Predicting Dynamic Stall," Paper Presented at the British Wind Energy Conference, Cambridge, U.K., April 1986.

65. Reddy, T.S.R., and Kaza, K.R.V., "A Comparative Study of Some Dynamic Stall Models," NASA TM-88917, March 1987.

Classical Structural Vibration:

66. Reissner, E., and Stein, M., "Torsion and Transverse Bending of Cantilever Plates," NACA TN-2369, June 1951.

67. Meirovitch, L., *Elements of Vibration Analysis*, McGraw-Hill, Inc., 1975.

Structural Vibration of Anisotropic Plates:

68. Crawley, E.F., "The Natural Modes of Graphite/Epoxy Cantilever Plates and Shells," *Journal of Composite Materials*, Vol. 13, July 1979, pp. 195-205.

69. Crawley, E.F., and Dugundji, J., "Frequency Determination and Non-Dimensionalization for Composite Cantilever Plates," *Journal of Sound and Vibration*, Vol. 72, No. 1, 1980, pp. 1-10.

70. Jensen, D.W., "Natural Vibrations of Cantilever Graphite/Epoxy Plates with Bending-Torsion Coupling," M.S. Thesis, Department of Aeronautics and Astronautics, M.I.T., August 1981.

71. Jensen, D.W., Crawley, E.F., and Dugundji, J., "Vibration of Cantilevered Graphite/Epoxy Plates with Bending-Torsion Coupling," *Journal of Reinforced Plastics and Composites*, Vol. 1, July 1982, pp. 254-269.

Finite Element & Finite Difference Schemes for Nonlinear Structures:

72. Bauchau, O.A., and Hong, C.-H., "Finite Element Approach to Rotor Blade Modelling," *Journal of the American Helicopter Society*, Vol. 32, No. 1, January 1987, pp. 60-67.

167

73. Hinnant, H.E., and Hodges, D.H., "Nonlinear Analysis of a Cantilever Beam," *AIAA Journal*, Vol. 26, No. 12, December 1988, pp. 1521-1527.

74. Minguet, P.J., "Static and Dynamic Behavior of Composite Helicopter Rotor Blades Under Large Deflections," Ph.D. Thesis, Department of Aeronautics & Astronautics, M.I.T., May 1989. (Also Technology Laboratory for Advanced Composites (TELAC) Rept. 89-7, M.I.T., May 1989.)

Ordering Schemes for Nonlinear Structures:

75. Hodges, D.H., and Dowell, E.H., "Nonlinear Equations of Motion for the Elastic Bending and Torsion of Twisted Nonuniform Rotor Blades," NASA TN D-7818, December 1974.

76. Hong, C.H., and Chopra, I., "Aeroelastic Stability of a Composite Rotor Blade," *Journal of the American Helicopter Society*, Vol. 30, No. 2, April 1985, pp. 57-67.

77. Rosen, A., and Friedmann, P.P., "The Nonlinear Behavior of Elastic Slender Beams Undergoing Small Strains and Moderate Rotations," *Trans. ASME, Journal of Applied Mechanics*, Vol. 46, No. 1, March 1979, pp. 161-168.

78. Boyd, W.N., "Effect of Chordwise Forces and Deformations and Deformation Due to Steady Lift on Wing Flutter," SUDAAR Rept. 508, Department of Aeronautics & Astronautics, Stanfoɪɔ University, December 1977.

Cubic Stiffening in Nonlinear Structures:

79. Tseng, W.Y., and Dugundji, J., "Nonlinear Vibrations of a Beam Under Harmonic Excitation," *Trans. ASME, Journal of Applied Mechanics*, Vol. 37, No. 2, Series E, June 1970, pp. 292-297.

80. Tseng, W.Y., and Dugundji, J., "Nonlinear Vibrations of a Buckled Beam Under Harmonic Excitation," *Trans. ASME, Journal of Applied Mechanics*, Vol. 38, No. 2, Series E, June 1971, pp. 467-476.

Stall Flutter Experiments:

81. Halfman, R.L., Johnson, H.C., and Haley, S.M., "Evaluation of High Angle-of-Attack Aerodynamic Derivative Data and Stall Flutter Prediction Techniques," NACA TN-2533, 1951.

82. Rainey, A.G., "Preliminary Study of Some Factors Which Affect the Stall-Flutter Characteristics of Thin Wings," NACA TN-3622, March 1956.

83. Rainey, A.G., "Measurement of Aerodynamic Forces for Various Mean Angles of Attack on an Airfoil Oscillating in Pitch and on Two Finite-Span Wings Oscillating in Bending with Emphasis on Damping in the Stall," NACA TR-1305, November 1957.

84. Dugundji, J., and Aravamudan, K., "Stall Flutter and Nonlinear Divergence of a 2-Dimensional Flat Plate Wing," Aeroelastic and Structures Research Laboratory TR 159-6, M.I.T., July 1974.

85. Dugundji, J., and Chopra, I., "Further Studies of Stall Flutter and Nonlinear Divergence of 2-Dimensional Wings," NASA CR-144924, August 1975. (Also, Aeroelastic and Structures Research Laboratory TR 180-1, August 1975.)

Composite Applications for Aeroelasticity:

86. Weisshaar, T.A., "Divergence of Forward Swept Composite Wings," *Journal of Aircraft*, Vol. 17, No. 6, June 1980, pp. 442-448.

87. Weisshaar, T.A., "Aeroelastic Tailoring of Forward Swept Composite Wings," *Journal of Aircraft*, Vol. 18, No. 8, August 1981, pp. 669-676.

88. Weisshaar, T.A., and Foist, B.L., "Vibration Tailoring of Advanced Composite Lifting Surfaces," *Journal of Aircraft*, Vol. 22, No. 2, February 1985, pp. 141-147.

89. Weisshaar, T.A., and Ryan, R.J., "Control of Aeroelastic Instabilities Through Stiffness Cross-Coupling," *Journal of Aircraft*, Vol. 23, No. 2, February 1986, pp. 148-155.

Flutter & Divergence with Composites:

90. Hollowell, S.J., and Dugundji, J., "Aeroelastic Flutter and Divergence of Stiffness Coupled, Graphite/Epoxy Cantilevered Plates," *Journal of Aircraft*, Vol. 21, No. 1, January 1984, pp. 69-76.

91. Selby, H.P., "Aeroelastic Flutter and Divergence of Rectangular Wings with Bending-Torsion Coupling," M.S. Thesis, Department of Aeronautics and Astronautics, M.I.T., January 1982.

92. Landsberger, B., and Dugundji, J., "Experimental Aeroelastic Behavior of Unswept and Forward Swept Graphite/Epoxy Wings," *Journal of Aircraft*, Vol. 22, No. 8, August 1985, pp. 679-686.

Harmonic Balance Methods for Nonlinear Flutter:

93. Kuo, C.-C., Morino, L., and Dugundji, J., "Perturbation and Harmonic Balance Methods for Nonlinear Panel Flutter," *AIAA Journal*, Vol. 10, No. 11, November 1972, pp. 1479-1484.

94. Dunn, P.E., "Stall Flutter of Graphite/Epoxy Wings with Bending-Torsion Coupling," M.S. Thesis, Department of Aeronautics & Astronautics, M.I.T., May 1989. (Also Technology Laboratory for Advanced Composites (TELAC) Rept. 89-5, M.I.T., May 1989.)

95. Dunn, P.E., and Dugundji, J., "Nonlinear Stall Flutter and Divergence Analysis of Cantilevered Graphite/Epoxy Wings," AIAA-90-0983, paper presented at the AIAA/ASME/ASCE/ AHS/ASC 31st Structures, Structural Dynamics and Materials Conference, Long Beach, California, April 1990.

TELAC Manufacturing Procedure:

96. Lagace, P.A., and Brewer, C.B., "TELAC Manufacturing Class Notes," Edition 0-2, Technology Laboratory for Advanced Composites, Department of Aeronautics and Astronautics, M.I.T.

Static Airfoil Data:

97. Jacobs, E.N., and Sherman, A., "Airfoil Section Characteristics as Affected by Variations of the Reynolds Number," NACA Report No. 586, 1937.

98. Jacobs, E.N., Ward, K.E., and Pinkerton, R.M., "The Characteristics of 78 Related Airfoil Sections from Tests in the Variable-Density Wind Tunnel," NACA Report No. 460, 1933.

171

172

Appendix A — Material Properties

The out-of-plane characteristics of graphite/epoxy laminates (i.e. the bending curvatures due to applied moments) have been observed to be experimentally different from in-plane characteristics (i.e. stretching due to applied extensional forces). These differences have also been observed to be layup and thickness dependent, although the thickness dependency may actually be due to manufacturing errors compounded by the z^3 factor in the D_{ij} terms, as hypothesized by Minguet [Ref. 74]. For the current investigation, there is no thickness dependency since all the laminates are of the same thickness. The layup dependency has been "smeared" across all the layups, so that in the current investigation only the out-of-plane bending moduli were used, no matter what the layup was.

	Hercules AS4/3501-6 Graphite/Epoxy	
	In-plane stretching	Out-of-plane bending
E_L, longitudinal modulus	143 GPa	97.3 GPa[*]
E_T, transverse modulus	9.7 GPa	6.3 GPa[*]
G_{LT}, shear modulus	4.9 GPa	5.3 GPa[**]
ν_{LT}, Poisson's ratio	0.30	0.28
ρ, density	1540 kg/m^3	1540 kg/m^3
t, ply thickness	0.135 mm	0.135 mm

[*] Based on static deflection tests
[**] Based on free vibration tests

The styrofoam properties were determined by averaging two tests. First, static deflection tests in bending and in torsion were performed on a piece of styrofoam 55 cm long and 12.7 cm by 1.6 cm in cross section. Next, vibration tests were performed on a smaller $[0_2/90]_S$ wing with styrofoam fairings [Refs. 94 and 95], and the styrofoam moduli adjusted until the analytic frequencies exactly matched the experimental frequencies. The styrofoam moduli were then assumed to be the average of these two values. The observed values from static deflection and free vibration were both within 25% of the final averaged values.

	HD-300 styrofoam	
	Nominal	Observed
E_L, longitudinal modulus	24 MPa	15 MPa***
E_T, transverse modulus	24 MPa	15 MPa***
G_{LT}, shear modulus	15 MPa	8 MPa***
ν_{LT}, Poisson's ratio	0.30	0.28
ρ, density	35 kg/m^3	35 kg/m^3
$Q_{11}^{sty} = Q_{22}^{sty}$	26.4 MPa	16.5 MPa
$Q_{12}^{sty} = Q_{21}^{sty}$	7.9 MPa	11.9 MPa
Q_{66}^{sty}	15.0 MPa	8.0 MPa

*** Average of static deflection and free vibration tests

Appendix B — Flat Plate Structural and Mode Shape Constants

	$[0_3/90]_S$	$[+15_2/0_2]_S$	$[-15_2/0_2]_S$
A_{11} (N/m)	8.09×10^7	9.96×10^7	9.96×10^7
A_{22} (N/m)	3.15×10^7	7.47×10^6	7.47×10^6
A_{12} (N/m)	1.91×10^6	4.59×10^6	4.59×10^6
A_{66} (N/m)	5.72×10^6	8.40×10^6	8.40×10^6
A_{16} (N/m)	-0-	1.08×10^7	-1.08×10^7
A_{26} (N/m)	-0-	1.53×10^6	-1.53×10^6
D_{11} (Nm)	10.1163	9.2478	9.2478
D_{22} (Nm)	0.8147	0.7718	0.7718
D_{12} (Nm)	0.1861	0.6418	0.6418
D_{66} (Nm)	0.5564	1.0121	1.0121
D_{16} (Nm)	-0-	1.8395	-1.8395
D_{26} (Nm)	-0-	0.2608	-0.2608

All B_{ij} values are zero because all layups are symmetric

Layup	β	n	g	f	B_{n1}	B_{n2}	B_{n4}
		1	1.8144	6.7476	-.24235	+.90076	-.24219
$[0_3/90]_S$	.02368	2	5.0358	8.2218	+.38956	-.63637	+.38977
		3	8.0742	10.365	-.42572	+.54295	-.42296
		1	1.7495	9.3334	-.17845	+.94938	-.17844
$[+15_2/0_2]_S$	.01190	2	5.0501	10.467	+.35933	-.73484	+.35454
		3	8.1550	12.270	-.39855	+.59909	-.39817
		1	1.7495	9.3334	-.17845	+.94938	-.17844
$[-15_2/0_2]_S$	.01190	2	5.0501	10.467	+.35933	-.73484	+.35454
		3	8.1550	12.270	-.39855	+.59909	-.39817

In all cases $B_{n3} = -B_{n1}$

176

Appendix C — Static Aerodynamic Models

Raw data for the static lift curve of the NACA 0012 airfoil is taken from Jacobs & Sherman [Ref. 97] and is empirically fit using the previously described division into polynomial regions. For the current study, the Reynolds number is very low, always below the critical Reynolds number of approximately 3.4×10^5. Therefore, no Reynolds number dependence was incorporated for varying free stream velocity. As illustrated in Fig. 33, the model of the 3-dimensional lift curve used in this study is divided into three regions and, for simplicity, each region is defined by a straight line: (i) below the stall angle, $\alpha_1 = 10^\circ$, the 3-dimensional lift slope is given by $a_{oL} = C_{L\alpha} = 0.8 * 5.9$ rad^{-1} (where the 0.8 factor comes from the finite-span correction for an aspect ratio of 8), (ii) between 10° and 20° the 3-dimensional lift coefficient drops linearly to 0.75, and (iii) above 20° the 3-dimensional lift coefficient remains constant at 0.75. The 3-dimensional moment coefficient follows the same trend: (i) it remains zero below the stall angle, (ii) drops linearly to -0.108 between 10° and 20°, and (iii) drops linearly to -0.150 between 20° and 37.5°. The two-dimensional profile drag is given by the polynomial,

$$(C-1) \qquad C_{D_o} = 4.923\alpha^3 + .1473\alpha^2 + .042\alpha + .014$$

Other 3-dimensional effects are included by adding a spanwise drop, as suggested by lifting line theory and approximated by a 9th order polynomial (see Landsberger [Ref. 92]). The 2-dimensional curves are already corrected for finite aspect ratio.

177
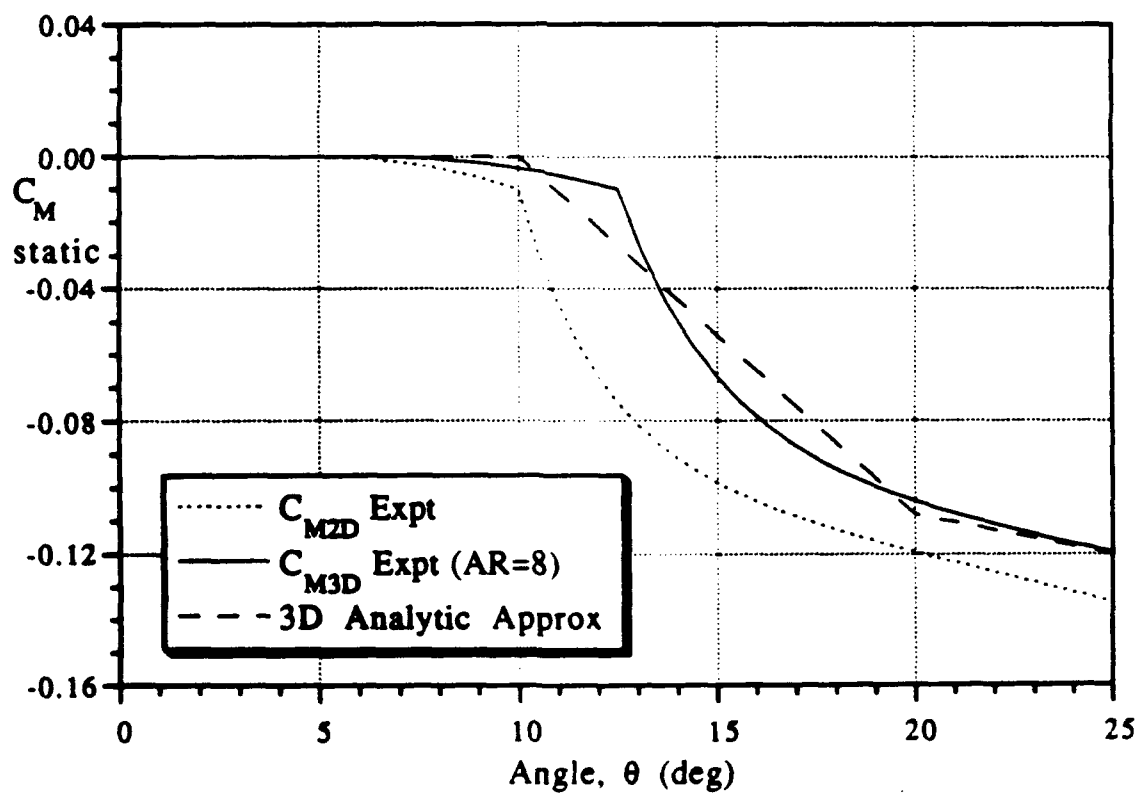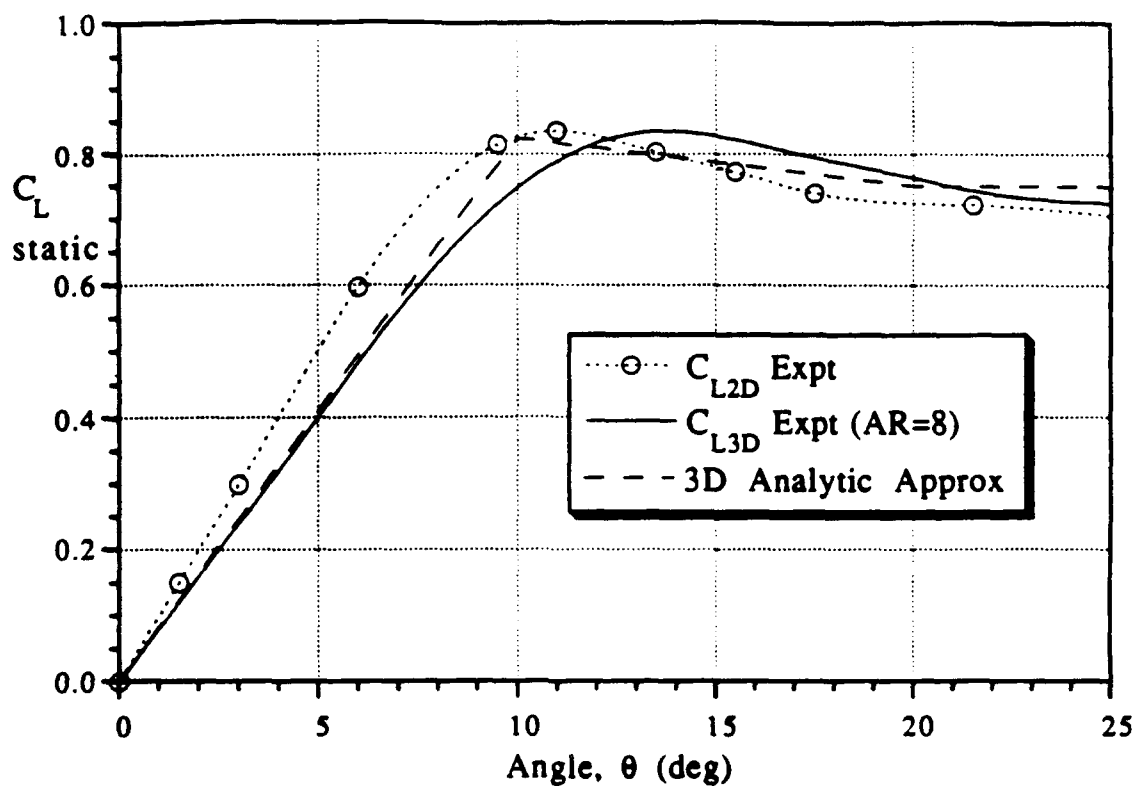
Fig. 33 NACA 0012 low Reynolds number lift model

178

$$\text{(C-2)} \qquad C_{L_{3D}} = 1.11 \left[1 - \left(\frac{x}{\ell}\right)^9 \right] C_{L_{2D}}(\alpha = \alpha_c)$$

where the corrected angle of attack included the finite-span correction, as suggested by Jacobs, Ward, & Pinkerton [Ref. 98],

$$\text{(C-3)} \qquad \alpha_c = \frac{1}{1 + \dfrac{a_{oL}}{\pi AR}} \alpha$$

The 3-dimensional total drag is found by adding the induced drag to the profile drag,

$$\text{(C-4)} \qquad C_D = C_{D_o} + \frac{C_L^2}{\pi AR}$$

As is suggested by Petot [Ref. 52], and illustrated in Fig. 34, more complex descriptions can be devised, and may be useful for higher Reynolds number flows where the lift drop after stall is more acute. A parabolic fit can be used to describe the slight drop in lift preceding stall. A power series expansion into a high order polynomial can be used to describe the exponential drop immediately following stall (the conversion from exponential form to polynomial form is necessitated by the formulation of the Fourier series in Section 3.2.2). A flat line can be used to describe the fully decayed exponential for very high angles of attack.

The variables describing the aerodynamic force curves, such as the maximum lift coefficient or the minimum profile drag, can further be generalized over a wide range of free stream velocities, as suggested by the logarithmic dependence on the Reynolds number

described by Jacobs & Sherman [Ref. 97]. Similar fits for the moment coefficient curve can be generated using the data from McAlister, Pucci, McCroskey, & Carr [Ref. 4].
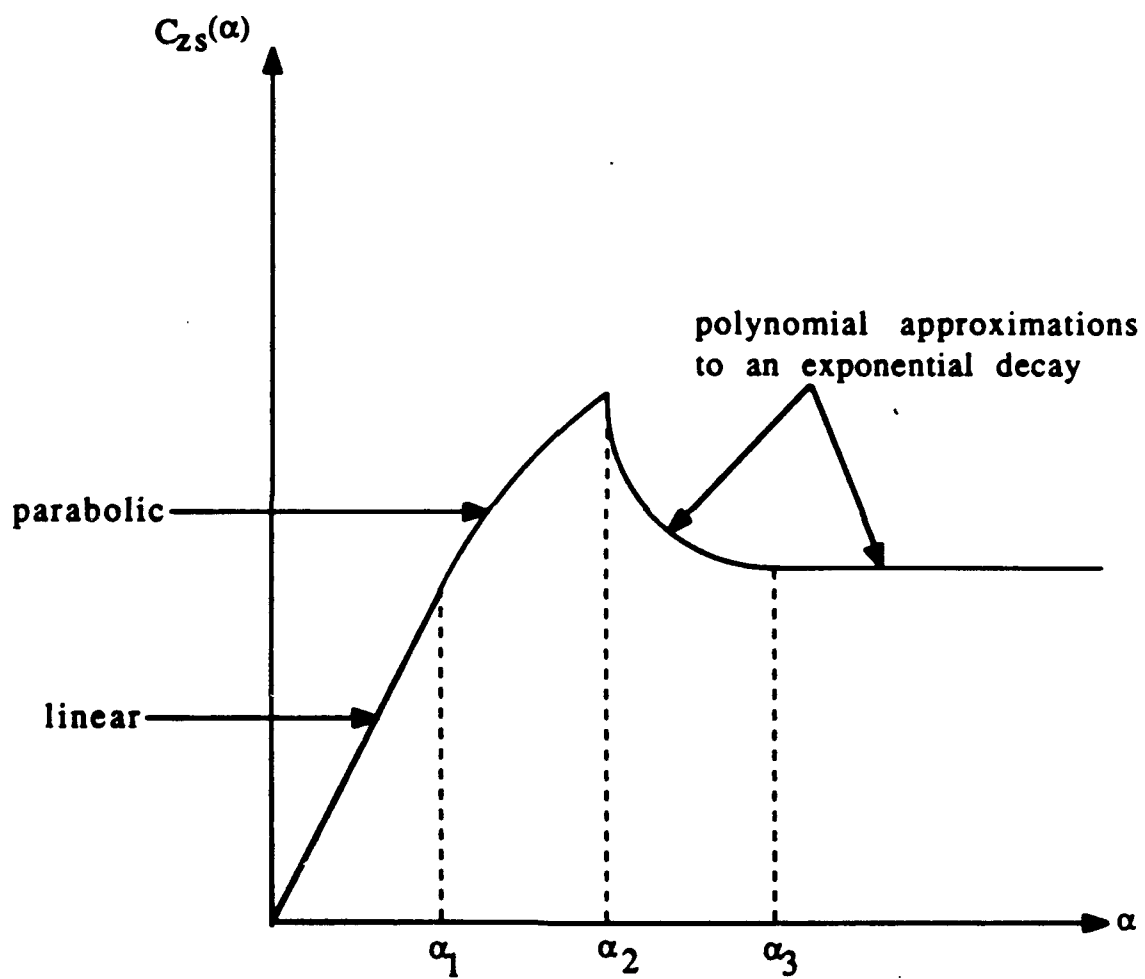
Fig. 34 Generalized lift model

Appendix D — Coefficients of Aerodynamic Equations

Table 5 shows the coefficients of the 2-dimensional aerodynamic equations (3-62) to (3-64), used for the lift and moment coefficients. It is assumed that there is no hysteresis in the drag coefficient. The linear coefficients (s_L, k_{vL}, λ_L, α_L, σ_L, a_{oM}, s_M, k_{vM}, λ_M, α_M, and σ_M) were taken from standard references with the following exceptions: s_L was taken from Petot [Ref. 52] although a more consistent value could have been $s_L = \pi$; a_{oL} was derived by fitting the NACA 0012 data from Jacobs & Sherman [Ref. 97] although the linear value $a_{oL} = 2\pi$ could have been used.

The nonlinear coefficients a and r for the NACA 0012 airfoil were taken from Petot [Ref. 52] for Reynolds numbers above the critical Reynolds number of 3.4×10^5. The nonlinear coefficient e for the NACA 0012 airfoil was taken from Petot & Dat [Ref. 54] for Reynolds numbers above the critical Re, since the form of the forcing terms used in Ref. 52 was unsuitable for determining an appropriate value of e, as is discussed in more detail later.

Corrections for low Reynolds flow were guided by similar values given by Petot & Loiseau [Ref. 51]. In that investigation, conducted for an OA 209 profile, a_o and r_o were determined to remain unchanged from high to low Reynolds number. In other words, it was determined that the characteristics of light stalling — i.e. when ΔC_z was small — were insensitive to Reynolds number.

However, it was also determined that the characteristics of moderate or deep stalling — i.e. when ΔC_z was no longer insignificant — were very sensitive to the Reynolds number. The value of a_1 rose

from 0.45 to 1.75 for the OA 209 airfoil, in other words by a factor of approximately 4. The form of the nonlinear coefficient r used in the study was,

$$(D-1) \qquad \sqrt{r} = r_0 + \alpha \Delta C_z + \beta + \frac{\beta^2}{\alpha \Delta C_z - \beta} \qquad \text{with } \beta = -1$$

$$\approx r_0 + \alpha \Delta C_z - 1 + (-1)^2 (1 - \alpha \Delta C_z + (\alpha \Delta C_z)^2)$$

$$\approx r_0 + \alpha^2 (\Delta C_z)^2$$

The above formulation indicates that r_1 corresponds approximately to α^2, for small values of ΔC_z. In Ref. 51, α rose from 0.65 to 1.0 for the OA 209 airfoil, indicating that the corresponding r_1 would rise from approximately 0.42 to 1.0, or by a factor of approximately 2.5.

The final value of the nonlinear coefficient e was governed by several influences. First, several forms of the forcing — or right hand side (RHS) — of the nonlinear ONERA equation were used in different studies,

$$(D-2) \qquad RHS = -[r \Delta C_z + e(\overset{\bullet}{\Delta C_z})]$$

$$(D-3) \qquad RHS = -[r \Delta C_z + e\overset{\bullet}{\theta}]$$

$$(D-4) \qquad RHS = -r[\Delta C_z + e(\overset{\bullet}{\Delta C_z})]$$

The form used in equation (D-2) is the form used by Petot & Loiseau [Ref. 51] for an OA 209 airfoil, in which it was determined that the value of e_1 rose from -0.6 to -2.7 going to low Reynolds number flow, in other words by a factor of approximately 4.5. Equation (D-2) is also the form used in the current investigation.

183

The form used in equation (D-3) is that used by Petot [Ref. 52]. While this form is easier to incorporate in a harmonic analysis such as the current investigation, it seems to be physically counter-intuitive since it indicates that there would be stalling influences due to $\dot{\theta}$ even in the unstalled region. For this reason the form of equation (D-3) was not used in the current investigation.

The form used in equation (D-4) is that used by Petot & Dat [Ref. 54] and seems to have a more sound physical basis: the phase-lag is expressed directly in relation to the force deficit ΔC_z, instead of in relation to $r\Delta C_z$ as in equation (D-2). However, this form is more difficult to implement in a harmonic balance method since the product of r and e would produce a sixth-order polynomial. Hence the form of equation (D-4) was not used in the current investigation. However, the value of the nonlinear coefficient e for a NACA 0012 airfoil above the critical Reynolds number *was* determined from Ref. 54 since Ref. 51, which uses the form of the current investigation, only looked at the OA 209 airfoil. The value of $e_1=-0.6$ was taken from Ref. 54 and multiplied by $r_0^2=.04$ to give the value of $e_1=-.024$ used in the current investigation.

In determining the nonlinear coefficient e, the effect of the fixed-time delay cannot be ignored. As mentioned previously, Petot & Loiseau [Ref. 51] determined that e_1 rose by a factor of approximately 4.5 for the OA 209 airfoil going to low Reynolds number flow. However, their investigation included a fixed-time delay while the current investigation does not. If the force deficit is approximated by only its first harmonic, i.e. using only Δ_0 and Δ_1, then the forcing term in the current investigation looks like,

184

(D-5) $RHS = -[r\Delta C_z + e(\Delta \overset{*}{C}_z)]$

$= -[r\Delta_0 + r\Delta_1 \sin(k\tau) + ek\Delta_1 \cos(k\tau) + H.H.T.]$

$= -[r\Delta_0 + \sqrt{r^2 + (ek)^2} \, \Delta_1 \sin(k(\tau + \Delta\tau))]$

$\approx -[r\Delta_0 + r\Delta_1 \sin(k(\tau + \Delta\tau))]$ for $ek \ll r$

with $\Delta\tau = \dfrac{1}{k} \tan^{-1}\dfrac{ek}{r} \approx \dfrac{e}{r}$ for $ek \ll r$

So, the nonlinear coefficient e can be interpreted as a fixed time delay parameter — however, it must be realized that this analogy breaks down for higher harmonics and larger values of ek. Also, to interpret the nonlinear coefficient as a fixed time delay parameter, it must be interpreted as affecting the entire forcing function as a whole, i.e. the entire right hand side of the stalled ONERA equation (as in Petot & Loiseau [Ref. 51]), and not simply the force deficit ΔC_z (as in Petot [Ref. 52] or Petot & Dat [Ref. 54]). That is, to interpret the nonlinear coefficient e as a fixed time delay parameter, it must be conceded that the nonlinear coefficients r & a are not affected by the time delay. While this assertion makes the mathematical formulation easier to interpret, it makes less physical sense, since intuitively the frequency and damping should be functions of the development of the stalling, and should therefore be affected by any delay which affects ΔC_z. In conclusion, a fixed time delay over a short period of the hysteresis cycle, such as used in Refs. 51-56, can be "smeared" over the entire cycle and eliminated from the mathematical formulation by appropriately adjusting the nonlinear coefficient e. It is for this reason that the high and low Reynolds number values

of e in Table 5 are so different, because $\Delta \tau$ has been left in for the former while it has been eliminated for the latter.

A sensitivity analysis was done on the flutter boundary of the $[0_3/90]_S$ wing to determine the final values of the nonlinear coefficients a, r, & e. The *initial* values of this sensitivity analysis were governed by the approximate correction factors that might be suggested by the investigation of Petot & Loiseau [Ref. 51]: a_0 remained constant at 0.25; a_1 rose by a factor of 4 from 0.1 to 0.4; r_0 remained constant at 0.2; r_1 rose by a factor of 2.5 from 0.1 to 0.25; e_1 rose by a factor of 4.5 from -.024 to -0.1. The sensitivity analysis attempted to find the smallest adjustments to these nonlinear values which could appropriately fit the $[0_3/90]_S$ flutter boundary.

The values of a_0 and a_1 had little effect on the flutter boundary, so they were left unchanged. The values of r_0 and r_1 were found to affect the flutter boundary, but had little effect on the flutter frequency plot, as shown in Fig. 35. Increasing r_0 increased the range over which the flutter boundary was governed by light stall. Increasing r_1 changed that part of the flutter boundary governed by deep stall such that the flutter velocity would decrease at a greater rate. Increasing e_1 had the same effect on the flutter boundary, but also increased the rate at which the frequency plot rose toward the first torsion frequency.

The procedure for making the final adjustments to the nonlinear coefficients was threefold: first, e_1 was increased from -.024 to +.030 so as to accurately fit the rise in the flutter frequency; second, r_1 was increased from 0.25 to 1.0 to account for the sharper drop in flutter velocity; third, r_0 was decreased from 0.2 to 0.1 to account for

186

the shorter range of light stall behavior. The changes in r_o and r_1 seem reasonable in comparison to the similarly large changes indicated by Petot & Loiseau [Ref. 51]. While at first the change in sign for e_1 seems odd, the "smearing" of $\Delta\tau$ is also included in this adjustment.

An alternative method of determining these coefficients of the nonlinear aerodynamics would have been to base them on the flutter boundary of the $[+15_2/0_2]_S$ instead. The reasoning would be that no matter what the root angle of attack α_R would be, most of the wing would be relatively untwisted, and so, except for 3-dimensional spanwise corrections, the aerodynamics across the span of the wing would be almost the same. The only problem with this approach is that the $[+15_2/0_2]_S$ layup has a very short light stall region, that is, it goes from the linear region to the deep stall region for only a small change in the root angle of attack (as can be seen in Fig. 25). Therefore, it would be difficult to determine the a_o, r_o, and e_o values from the $[+15_2/0_2]_S$ layup.

	Lift	Moment
s_z	$0.09*(180/\pi)$ rad^{-1}	$-\pi/4$ rad^{-1}
k_{vz}	$\pi/2$ rad^{-1}	$-3\pi/16$ rad^{-1}
λ_z	0.15	0
α_z	0.55	1
σ_z	5.9 rad^{-1}	$-\pi/4$ rad^{-1}
a^*	$0.25 + 0.1(\Delta C_L)^2$	
r^*	$[0.2 + 0.1(\Delta C_L)^2]^2$	
e^*	$-.024(\Delta C_L)^2$	
$\Delta\tau^*$	10	
a^{**}	$0.25 + 0.4(\Delta C_L)^2$	
r^{**}	$[0.1 + 1.0(\Delta C_L)^2]^2$	
e^{**}	$+.030(\Delta C_L)^2$	
$\Delta\tau^{**}$	0	

* Re $> 3.4\times10^5$
** Re $< 3.4\times10^5$

Table 5. Aerodynamic Coefficients

188
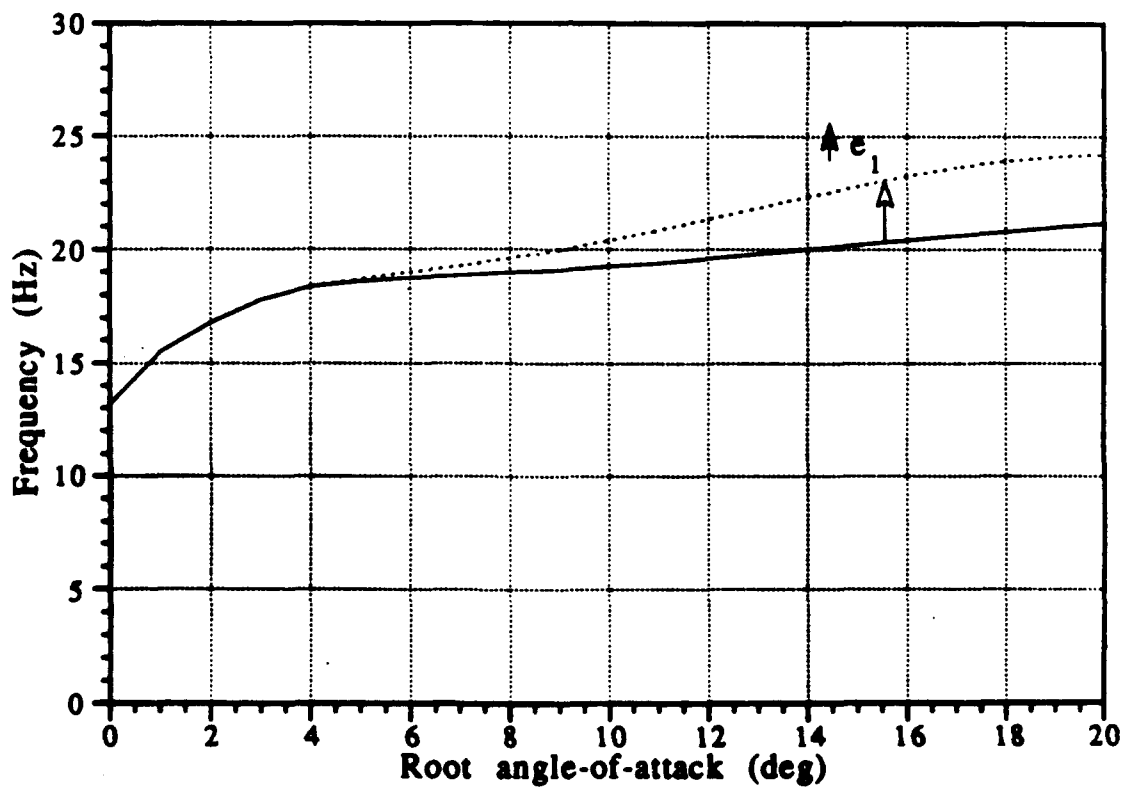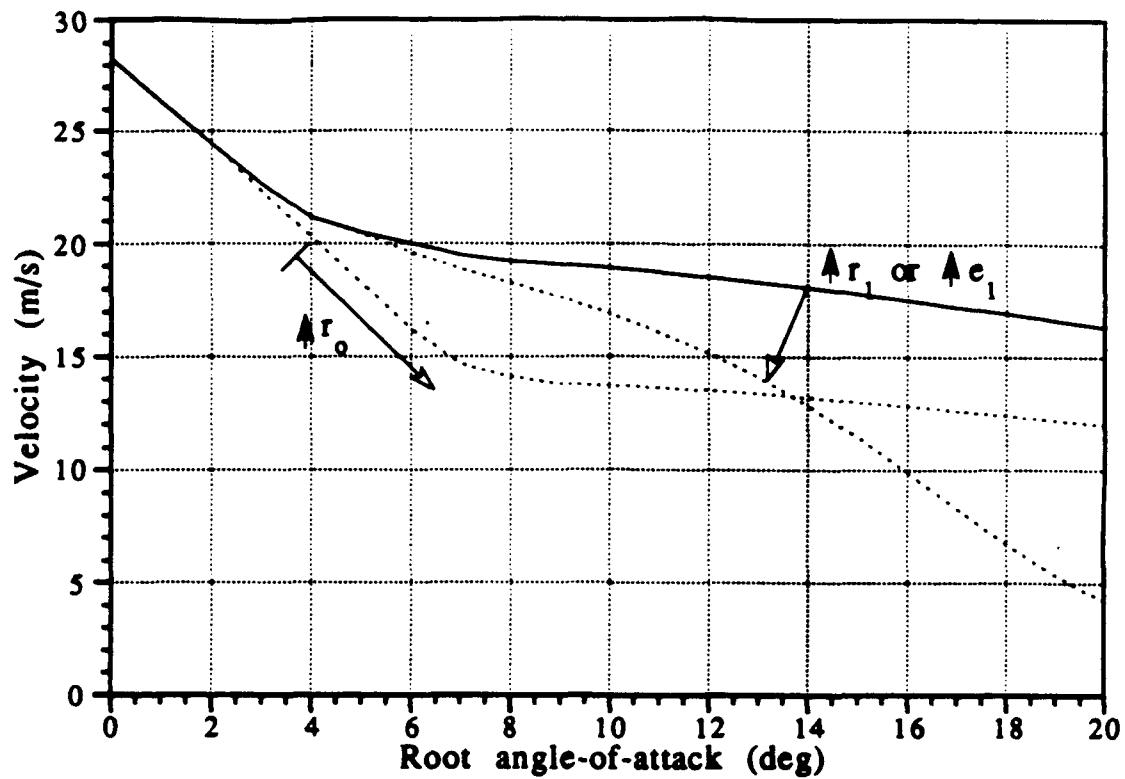
Fig. 35 Nonlinear coefficient sensitivity analysis on $[0_3/90]_S$ flutter boundary

Appendix E — Example of Fourier Analysis

Equations (3-78) to (3-80) are still applicable:

(E-1) $\qquad \alpha(\tau) = \alpha_0 + \alpha_v \sin(k\tau + \xi)$

where,

(E-2) $\qquad \alpha_v = \sqrt{\alpha_s^2 + \alpha_c^2}$

(E-3) $\qquad \xi = \tan^{-1} \dfrac{\alpha_c}{\alpha_s}$

For a single break point model (see Fig. 36), equation (3-69) simplifies to the following equations, where α_Δ is the stall angle and a_{11} is the difference in slopes between the linear region and the non-linear region,

(E-4) $\qquad \Delta C_z = \begin{cases} a_{11}(\alpha - \alpha_\Delta) & \text{for } \alpha \geq \alpha_\Delta \\ 0 & \text{for } \alpha \leq \alpha_\Delta \end{cases}$

Equations (3-84) to (3-87) then give that,

(E-5) $\qquad b_{10} = a_{11}(\alpha_0 - \alpha_\Delta)$

(E-6) $\qquad b_{11} = a_{11}\alpha_v$

(E-7) $\qquad I_{10} = \dfrac{\pi}{2} - \varphi_\Delta$

(E-8) $\qquad I_{11} = \cos\varphi_\Delta$

190

(E-9) $\qquad I_{12} = \frac{1}{2}\sin\varphi_\Delta\cos\varphi_\Delta + \frac{\pi}{4} - \frac{1}{2}\varphi_\Delta$

where equation (3-86) is,

(E-10) $\qquad \varphi_\Delta = \begin{cases} \sin^{-1}\left\{\dfrac{\alpha_\Delta-\alpha_0}{\alpha_v}\right\} \\[2em] +\dfrac{\pi}{2} \qquad \text{if } \dfrac{\alpha_\Delta-\alpha_0}{\alpha_v} > +1 \\[2em] -\dfrac{\pi}{2} \qquad \text{if } \dfrac{\alpha_\Delta-\alpha_0}{\alpha_v} < -1 \end{cases}$

Finally, putting these expressions into the combined mean and oscillatory components of the nonlinear aerodynamic deviations (3-83) and (3-90), we get,

(E-11) $\qquad \Delta C_{zo} = \frac{a_{11}\alpha_v}{\pi}\left\{ -\frac{\alpha_\Delta-\alpha_0}{\alpha_v}\left[\frac{\pi}{2} - \varphi_\Delta\right] + \cos\varphi_\Delta\right\}$

(E-12) $\qquad \Delta C_{zv} = \frac{a_{11}\alpha_v}{\pi}\left\{ -\sin\varphi_\Delta \cos\varphi_\Delta + \frac{\pi}{2} - \varphi_\Delta\right\}$

A symmetric aerodynamic force curve can also be accounted for by including a second stall angle at $-\alpha_\Delta$. This yields expanded versions of Equations (E-11) and (E-12),

(E-13) $\qquad \Delta C_{zo} = \frac{a_{11}\alpha_v}{\pi}\left\{ -\frac{\alpha_\Delta-\alpha_0}{\alpha_v}\left[\frac{\pi}{2} - \varphi_\Delta\right] + \cos\varphi_\Delta\right\}$

$\qquad\qquad\qquad + \frac{a_{11}\alpha_v}{\pi}\left\{ -\frac{-\alpha_\Delta-\alpha_0}{\alpha_v}\left[\frac{\pi}{2} + \bar\varphi_\Delta\right] - \cos\bar\varphi_\Delta\right\}$

191

$$(\text{E-14}) \qquad \Delta C_{zv} = \frac{a_{11}\alpha_v}{\pi}\left\{-\sin\varphi_\Delta \, \cos\varphi_\Delta + \frac{\pi}{2} - \varphi_\Delta\right\}$$

$$+ \frac{a_{11}\alpha_v}{\pi}\left\{+\sin\bar{\varphi}_\Delta \, \cos\bar{\varphi}_\Delta + \frac{\pi}{2} + \bar{\varphi}_\Delta\right\}$$

where the additional form of equation (3-86) that is required is,

$$(\text{E-15}) \qquad \bar{\varphi}_\Delta = \begin{cases} \sin^{-1}\left\{\dfrac{-\alpha_\Delta - \alpha_0}{\alpha_v}\right\} \\[3ex] +\dfrac{\pi}{2} \qquad \text{if } \dfrac{-\alpha_\Delta - \alpha_0}{\alpha_v} > +1 \\[3ex] -\dfrac{\pi}{2} \qquad \text{if } \dfrac{-\alpha_\Delta - \alpha_0}{\alpha_v} < -1 \end{cases}$$

192
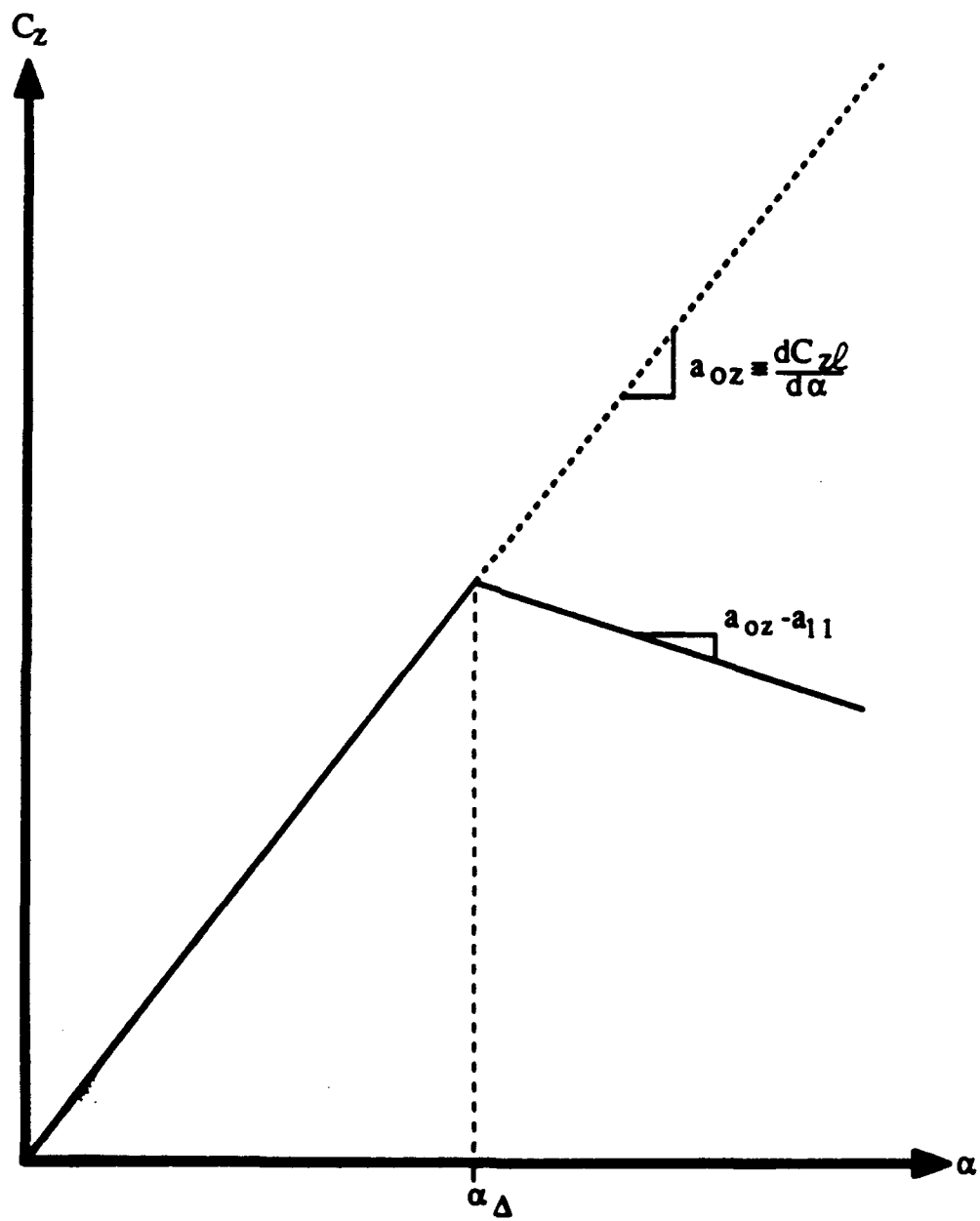
Fig. 36 Example of single break point stall model

Appendix F — The Newton-Raphson Method

The Newton-Raphson method is a numerical solver used to find the roots of the implicit vector equation, $f(x) = 0$, where x is the state vector and $f(x)$ is the vector of residual functions that must be driven to zero. The Newton-Raphson scheme takes an initial guess of the state vector x and drives the vector $f(x)$ toward zero by inverting the Jacobian matrix (derivative matrix), and obtaining a correction Δx to the current guess. The process is repeated until the correction Δx becomes negligible and the process is deemed to have converged.

$$(F-1) \qquad \Delta x^{(n)} = -\left[\frac{df}{dx}\right]_n^{-1} f(x^{(n)}) \quad ; \quad x^{(n+1)} = x^{(n)} + \Delta x^{(n)}$$

The Newton-Raphson solver is applied in the current analysis by rearranging equation (3-217) as follows;

$$(F-2) \qquad \{f\} = \begin{bmatrix} [K] & 0 & 0 \\ 0 & -\omega^2[M]+[K] & 0 \\ 0 & 0 & -\omega^2[M]+[K] \end{bmatrix} \begin{bmatrix} \{q_o\} \\ \{q_s\} \\ \{q_c\} \end{bmatrix} - \begin{bmatrix} \{Q_o\} \\ \{Q_s\} \\ \{Q_c\} \end{bmatrix}$$

Equation (F-2) comprises 3n equations that must be solved in the form $f(x) = 0$ and are nonlinear in the aerodynamic dependence of the modal forces Q_i on the modal amplitudes q_i and in the structural dependence of the stiffness matrix K_{ij} on the modal amplitudes q_i. The state vector x is comprised of the harmonic components of the modal amplitudes, q_{io}, q_{is}, and q_{ic}, with some minor adjustments to ensure convergence to a non-trivial solution: the sine component

194

of one mode q_i is set to some small constant to set the amplitude level, while its cosine component is set to zero, since the flutter limit cycle oscillations can start at any arbitrary phase. The mode usually chosen for this substitution is the first torsional mode, since experimentally it is this mode which dominates the oscillatory motion. These sine and cosine components are then dropped from the state vector x and are replaced by the reduced frequency of oscillation, k, and the flutter velocity, U. Because the sinusoidal component of one mode shape has already been set to a non-zero value, the Newton-Raphson scheme does not converge to the trivial steady solution.

Note that the Newton-Raphson solver is not always guaranteed to converge, especially when the initial guess is too far from the ultimate solution or when the derivatives used in the Jacobian matrix are changing abruptly, which often happens with nonlinearities that have discontinuous derivatives. In regions where convergence is difficult (for example near the stall angle where the lift/moment coefficient curves are discontinuous in slope), a relaxation technique, which consists of taking only a fraction of Δx as a correction for each iteration, is more likely to converge. When the Newton-Raphson solver does converge to a solution, it will satisfy the equations, but there is no indication as to whether this solution is unique or not. If other solutions exist, the only way to find them is to start with a different initial state vector.

The Jacobian matrix can be calculated either numerically or analytically. The numerical method involves moving an incremental distance in each direction of the state vector x, finding the resulting incremental change in the residual vector f, and estimating each

195

component of the derivative matrix as $\Delta f_i / \Delta x_j$. The analytic method involves carrying out the entire nonlinear differentiation, which is best carried out by multiple application of the chain rule to the equations of Chapter 3.

Both the numerical and analytic methods have their advantages and disadvantages. The numerical method is easier to code on a computer since it involves using the already existing subroutines which must compute the residual vector. On the other hand, the numerical method is computationally inefficient since it requires recalculating the residual vector for every direction of the state vector. In addition, the numerical method is likely to be inaccurate at points of discontinuity in derivatives, unless the user is careful to choose appropriately small increments in the state vector.

By contrast, the analytic model directly solves for the Jacobian matrix without needing several iterations, and so is computationally faster for higher numbers of mode shapes and harmonics. In addition, the analytic method is always accurate and does not depend on any step size. Unfortunately, the analytic method cannot employ already existing subroutines and requires cumbersome programming for a highly nonlinear problem such as in the current study.

Both the numerical and the analytic methods were used to calculate the Jacobian matrix in the current study and compared well against each other. However, with the large complexity in calculating the modal forces, the numerical method is more likely to have fewer coding errors than the analytic method, despite being computationally slower. For this reason, it was used more extensively for the full flutter analyses.

196

Appendix G — Computer Code

```fortran
C-----FILE : FLUTTER.FOR------------------------------------------------
C
      PROGRAM FLUTTER
C
      INCLUDE     PARAM.INC
      INCLUDE     GLBBLK.INC
      REAL        QLIT(MAXMODE,3),QDUM(MAXMODE,3)
      REAL        QALL(3*MAXMODE),RES(3*MAXMODE),QBIG(MAXMODE,3)
      REAL        DRDQ(3*MAXMODE,3*MAXMODE),DQALL(3*MAXMODE)
      REAL        VEL,AOA,FREQ,ATIP(3),HTIP(3),VTIP(3)
      REAL        SMAX,RMAX
      INTEGER     LMAX,IERR,BEN_TOR,MBT
      LOGICAL     CONVERGED,LNEWT,LSTRUC,LCUBE,LFLUTB,LSTART
      CHARACTER   LAYUP*25,ANSWER*1,FILENAME*25,CDUM*8
C
      REAL        A0L,A1L,R0L,R1L,E0L,E1L
      REAL        A0M,A1M,R0M,R1M,E0M,E1M
      COMMON      / COEFBLK / A0L,A1L,R0L,R1L,E0L,E1L,
     &            A0M,A1M,R0M,R1M,E0M,E1M
C
C     QLIT(i,j):  i-th modal amplitude, j-th component (1=mean,
C                 2=sine,3=cosine)
C     QALL:       Augmented state vector
C     RES:        Residual vector
C     DRDQ:       Jacobian matrix, derivatives of residuals (RES)
C                 w.r.t. the state vector (QALL)
C     DQALL:      Corrections to augmented state vector
C     VEL:        Free stream velocity
C     AOA:        Root angle-of-attack
C     FREQ:       Reduced frequency
C     ATIP:       Components of oscillating tip angle
C     HTIP:       Components of oscillating tip deflection
C     IERR:       Error status variable for opening of data file
C     BEN_TOR:    Integer variable denoting whether analysis assumes
C                 bending (BEN_TOR=1) or torsional (BEN_TOR=2) flutter
C     CONVERGED:  Logical variable to tell if Newton-Raphson solver
C                 has converged to a solution
C     LNEWT:      Logical variable to tell if diagnostics are to be
C                 printed to output file at each step of the N-R solver
C     LSTRU:      Logical variable to tell if diagnostics are to be
C                 printed to output file on structural variables
C     LAYUP:      Character variable to denote flat plate layup (eg. for
C                 [0:2/90]:s, LAYUP might be '0290'). All data files
C                 must be of the form <LAYUP>.DAT.
C
      FOIL = 'NAC12'
C
C     Read in the layup.
C
10    WRITE(*,'(A,$)') ' Layup : '
      READ(*,'(A)',ERR=10) LAYUP
20    WRITE(*,'(A,$)') ' Newton-Raphson control file : '
      READ(*,'(A)',ERR=20) FILENAME
C
C     Read in specifications of current run from control file.
C
30    FORMAT(/8X,5I8///8X,7L8///8X,I8,2L8,I8,2G8.0,I8)
      OPEN(UNIT=2,FILE=TRIM(FILENAME),
     & STATUS='OLD',FORM='FORMATTED',ERR=20)
      READ(2,30,ERR=20) NB,NT,NC,NF,BEN_TOR,LATAN,LCUBE,
```

197

```fortran
      & LINEAR,CORREC,REDUC,STEADY,VLINES,ATYPE,LCONST,
      & LGEOM,IGEOM,SMAX,RMAX,LMAX
        CLOSE(2)
        NMODES = NB+NT+NC+NF
        IF ((NB.LT.1).OR.(NB.GT.NBMAX)) GOTO 20
        IF ((NT.LT.1).OR.(NT.GT.NTMAX)) GOTO 20
        IF ((NC.LT.0).OR.(NC.GT.NCMAX)) GOTO 20
        IF ((NF.LT.0).OR.(NF.GT.NFMAX)) GOTO 20
C
C     Read in nonlinear aerodynamic coefficients.
C
        IF ((ATYPE.NE.0).AND.(ATYPE.NE.1)) THEN
35        WRITE(*,'(A,$)') ' Aero control file : '
          READ(*,'(A)',ERR=35) FILENAME
          OPEN(UNIT=2,FILE=TRIM(FILENAME),STATUS='OLD',
      &     FORM='FORMATTED',ERR=35)
          READ(2,'(/8X,6G8.0///8X,6G8.0)',ERR=35) A0L,A1L,
      &     R0L,R1L,E0L,E1L,A0M,A1M,R0M,R1M,E0M,E1M
          CLOSE(2)
        ENDIF
C
        LNEWT = .FALSE.
40      WRITE(*,'(/A,$)') ' Output Newton-Raphson troubleshooting '//
      & 'diagnostics ? '
        READ(*,'(A)',ERR=40) ANSWER
        IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) LNEWT = .TRUE.
C
        LAEROF = .FALSE.
50      WRITE(*,'(A,$)') ' Output aerodynamics troubleshooting '//
      & 'diagnostics ? '
        READ(*,'(A)',ERR=50) ANSWER
        IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) LAEROF = .TRUE.
C
        LSTRUC = .FALSE.
55      WRITE(*,'(A,$)') ' Output structures troubleshooting '//
      & 'diagnostics ? '
        READ(*,'(A)',ERR=55) ANSWER
        IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) LSTRUC = .TRUE.
C
        LSTART = .FALSE.
60      WRITE(*,'(A,$)') ' Write restart file ? '
        READ(*,'(A)',ERR=60) ANSWER
        IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) THEN
          LSTART = .TRUE.
          WRITE(FILENAME,'(A,4I1)') TRIM(LAYUP)//
      &     '.start',NB,NT,NC,NF
          OPEN(UNIT=3,FILE=TRIM(FILENAME),STATUS='NEW',
      &     FORM='FORMATTED',ERR=60)
        ENDIF
C
        DO 65 I = 1,NMODES
          IF (I.LE.NB) WRITE(MLABEL(I),'(A,I1)') 'B',I
          IF ((I.GT.NB).AND.(I.LE.NB+NT))
      &     WRITE(MLABEL(I),'(A,I1)') 'T',I-NB
          IF ((I.GT.NB+NT).AND.(I.LE.NB+NT+NC))
      &     WRITE(MLABEL(I),'(A,I1)') 'C',I-(NB+NT)
          IF ((I.GT.NB+NT+NC).AND.(I.LE.NB+NT+NC+NF))
      &     WRITE(MLABEL(I),'(A,I1)') 'F',I-(NB+NT+NC)
65      CONTINUE
C
C     Create mass and stiffness matrices by calling STATIC subroutine.
C
        CALL STATIC(LAYUP,LSTRUC,TRATIO,IERR)
        IF (IERR.NE.0) THEN
```

198

```fortran
          WRITE(*,'(A,I2,A)') ' IOSTAT=',IERR,' error reading '//
     &      TRIM(LAYUP)//'.DAT data file.'
          GOTO 10
        ENDIF
C
C     Open output file.
C
        FILENAME = TRIM(LAYUP)//'WNAV.OUT '
        OPEN(UNIT=2,FILE=TRIM(FILENAME),
     & STATUS='NEW',FORM='FORMATTED',IOSTAT=IERR)
        IF (IERR.EQ.0) THEN
          WRITE(*,*) ' Analysis results being sent to ',
     &      TRIM(FILENAME)
        ELSE
          WRITE(*,'(A,I2,A)') ' IOSTAT=',IERR,' error opening '//
     &      TRIM(FILENAME)//' as output file.'
          GOTO 10
        ENDIF
C
        WRITE(2,'(4(/I2,A))') NB,' = number of bending modes',
     & NT,' = number of torsion modes',
     & NC,' = number of chordwise bending modes',
     & NF,' = number of fore-&-aft modes'
        WRITE(2,'(F4.2,A)') TRATIO,' = NACA airfoil thickness ratio'
C
        WRITE(2,*) ' Exact angle calculation =',LATAN
        WRITE(2,*) ' Cubic stiffening =',LCUBE
        IF (.NOT.LCUBE) KTTCUBE = 0.
        WRITE(2,*) ' Linear aerodynamics =',LINEAR
        WRITE(2,*) ' Spanwise lift correction =',CORREC
        WRITE(2,*) ' Finite span lift reduction =',REDUC
        WRITE(2,*) ' Steady test case (no flutter) =',STEADY
        IF ((.NOT.STEADY).AND.((ATYPE.EQ.0).OR.(ATYPE.EQ.1)))
     & WRITE(2,*) ' Unsteady analysis type =',ATYPE,
     & ' (see listing of COEFS.FOR)'
        IF (.NOT.STEADY) WRITE(2,*) ' Constant coeffs in '//
     & 'unsteady analysis =',LCONST
        WRITE(2,*) ' Geometric structural nonlinearities =',LGEOM
        WRITE(2,'(A,1PE6.0E1)') ' N-R max allowable step size =',SMAX
        WRITE(2,'(A,1PE6.0E1)') ' N-R max allowable residual  =',RMAX
        WRITE(2,'(A,I5)') ' Max number of iterations =',LMAX
C
C     Read in the start & end values and the incremental step size
C     between each line of either (i) constant velocity or
C     (ii) constant root angle of attack.
C
70      IF (VLINES) WRITE(*,'(A,$)') ' Velocity start, end, '//
     & '& step size (m/s) ? '
        IF (.NOT.VLINES) WRITE(*,'(A,$)') ' Root angle start, '//
     & 'end, & step size (deg) ? '
        READ(*,*,ERR=70) DUM1LO,DUM1HI,DUM1INC
        IF ((DUM1LO.LT.0.).OR.(DUM1HI.LT.0.)) GOTO 70
        IF (((DUM1HI-DUM1LO)/DUM1INC).LT.0.) GOTO 70
C
C     Write header.
C
        IF (.NOT.LNEWT) THEN
          IF (STEADY) THEN
            WRITE(2,'(/A,$)') '  Vel    AOA      H avg    A avg'
            IF (NF.EQ.0) WRITE(2,'(A)') ' '
            IF (NF.GT.0) WRITE(2,'(A)') '    V avg'
            WRITE(2,'(A,$)') '  (m/s)  (deg)     (cm)     (deg)'
            IF (NF.EQ.0) WRITE(2,'(A)') ' '
            IF (NF.GT.0) WRITE(2,'(A)') '      (cm)'
```

199

```
            ELSEIF ((.NOT.STEADY).AND.(BEN_TOR.EQ.1)) THEN
                WRITE(2,'(/A,$)')  '    Vel    AOA    Freq'//
     &          '     H avg    H amp    A avg    A amp    A phz'
                IF (NF.EQ.0) WRITE(2,*) ' '
                IF (NF.GT.0) WRITE(2,*) '    V avg    V amp    V phz'
                WRITE(2,'(A,$)')   '    (m/s)   (deg)    (Hz)'//
     &          '     (cm)     (cm)    (deg)    (deg)    (deg)'
                IF (NF.EQ.0) WRITE(2,*) ' '
                IF (NF.GT.0) WRITE(2,*) '     (cm)     (cm)    (deg)'
            ELSEIF ((.NOT.STEADY).AND.(BEN_TOR.EQ.2)) THEN
                WRITE(2,'(/A,$)')  '    Vel    AOA    Freq'//
     &          '     H avg    H amp    H phz    A avg    A amp'
                IF (NF.EQ.0) WRITE(2,*) ' '
                IF (NF.GT.0) WRITE(2,*) '    V avg    V amp    V phz'
                WRITE(2,'(A,$)')   '    (m/s)   (deg)    (Hz)'//
     &          '     (cm)     (cm)    (deg)    (deg)    (deg)'
                IF (NF.EQ.0) WRITE(2,*) ' '
                IF (NF.GT.0) WRITE(2,*) '     (cm)     (cm)    (deg)'
            ENDIF
        ENDIF
C
C       Initialize state vectors.
C
        DO 80 I = 1,MAXMODE
        DO 80 J = 1,3
            QLIT(I,J) = 0.
            QALL((I-1)*3+J) = 0.
80      CONTINUE
C
        LFLUTB = .FALSE.
        IF ((.NOT.STEADY).AND.(DUM1LO.NE.DUM1HI)) THEN
90          WRITE(*,'(A,$)') ' Flutter boundary analysis ? '
            READ(*,'(A)',ERR=90) ANSWER
            IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) LFLUTB = .TRUE.
        ENDIF
C
C       Loop through each line of either (i) constant velocity or
C       (ii) constant root angle of attack, denoted by the dummy
C       variable DUMMY1.
C
        DO 999 DUMMY1 = DUM1LO,DUM1HI,DUM1INC
            IF (STEADY) THEN
                FREQ = 0.
                INCLUDE STEADY.INC
            ELSE
                INCLUDE UNSTEADY.INC
            ENDIF
999     CONTINUE
        CLOSE(2)
        IF (LSTART) CLOSE(3)
        STOP
        END

C-----FILE: STEADY.INC --------------------------------------------------
C
        RGEOM = 1.
C
C       Set the velocity VEL or the root angle of attack AOA, depending
C       on whether lines of constant velocity or constant angle.
C
        IF (VLINES) VEL = DUMMY1
        IF (.NOT.VLINES) AOA = DUMMY1*PI/180.
C
C       Initialize to zero the augmented modal amplitude vector QALL,
```

200

```
C       and all the modal amplitudes QLIT.
C
        IF (DUMMY1.EQ.DUM1LO) THEN
            DO 1010 I = 1,NMODES
                QLIT(I,1) = 0.
                QALL(I) = QLIT(I,1)
1010        CONTINUE
        ENDIF
C
C       If steady, read in the start & end values and the incremental
C       step size of the root angles/velocities for each corresponding
C       line of constant velocity/root angle.
C
1020    IF (VLINES) THEN
            WRITE(*,'(A,F6.2,A,$)') ' VEL =',DUMMY1,
     &        ' ; Root angle start, end, & step size (deg) ? '
        ELSE
            WRITE(*,'(A,F6.2,A,$)') ' AOA =',DUMMY1,
     &        ' ; Velocity start, end, & step size (m/s) ? '
        ENDIF
        READ(*,*,ERR=1020) DUM2LO,DUM2HI,DUM2INC
        IF ((.NOT.VLINES).AND.(DUM2LO.LT.0.)) GOTO 1020
        IF ((.NOT.VLINES).AND.(DUM2HI.LT.0.)) GOTO 1020
        IF ((DUM2HI-DUM2LO)/DUM2INC.LT.0.) GOTO 1020
C
C       Read in the non-dimensional step size relaxation factor to
C       be applied to the corrections in the Newton-Raphson solver.
C
        FACTOR = 1.
        IF ((DUMMY1.EQ.DUM1LO).AND.(LGEOM)) THEN
1025        WRITE(*,'(A,$)') ' Relaxation factor ? '
            READ(*,*,ERR=1025) FACTOR
            IF ((FACTOR.LE.0.).OR.(FACTOR.GT.1.)) GOTO 1025
        ENDIF
C
C       Loop through the appropriate variable, denoted by the dummy
C       variable DUMMY2, for each line of constant velocity/root angle.
C
        DO 1999 DUMMY2 = DUM2LO,DUM2HI,DUM2INC
C
C           Initialize the number of iterations to zero and extract the
C           appropriate root angle/velocity from the dummy variable DUMMY2.
C
            LOOPS = 0
            IF (VLINES) AOA = DUMMY2*PI/180.
            IF (.NOT.VLINES) VEL = DUMMY2
C
C           Loop through the Newton-Raphson scheme until it is
C           converged to an acceptable limit.
C
            CONVERGED = .FALSE.
            DO WHILE (.NOT.CONVERGED)
C
C               Extract the modal amplitudes from
C               the augmented modal amplitude vector.
C
                DO 1030 I = 1,NMODES
                    QLIT(I,1) = QALL(I)
1030            CONTINUE
                LOOPS = LOOPS+1
C
C               Write current values of inputs to residual calculations.
C
1040            FORMAT (/A7,I5,12X,13(4X,A2,4X))
```

201

```
1045            FORMAT (A23,13(1PE10.2))
                IF (LNEWT) THEN
                    WRITE(2,1040) ' LOOP =',LOOPS,
     &              (MLABEL(I),I=1,NMODES)
                    WRITE(2,1045) ' Avg modal amp [m] : ',
     &              (QLIT(I,1),I=1,NMODES)
                ENDIF
C
C               Calculate the residuals from subroutine RESIDUAL, which
C               are functions of the velocity VEL, root angle of attack AOA,
C               reduced frequency FREQ, and modal amplitudes QLIT.
C
                CALL RESIDUAL(VEL,AOA,FREQ,QLIT,RGEOM,RES,QBIG)
                IF (LNEWT) WRITE(2,1045) ' Avg modal Qs  (ND): ',
     &          (QBIG(I,1),I=1,NMODES)
                IF (LNEWT) WRITE(2,1045) ' Avg residuals (ND): ',
     &          (RES(I),I=1,NMODES)
C
C               Calculate the derivative matrix of the residuals wrt the
C               modal amplitudes using subroutine R_REDIV, which is a
C               function of the velocity VEL, root angle of attack AOA,
C               reduced frequency FREQ, and modal amplitudes QLIT. The
C               current values of the residuals RES are also passed since
C               the derivative matrix may be calculated numerically, in
C               which case the current values are needed.
C
                CALL R_DERIV(BEN_TOR,VEL,AOA,FREQ,QLIT,RES,RGEOM,DRDQ)
C
C               Write derivative matrix.
C
                IF (LNEWT) THEN
                    WRITE(2,'(/A)') ' NUMERIC dR/dq MATRIX :'
                    WRITE(2,'(12X,13(6X,I1,3X))') (I,I=1,NMODES)
                    DO 1050 J = 1,NMODES
                        WRITE(2,'(4X,A8,13(1PE10.2))')
     &                  'dR/dq'//MLABEL(J)//'o',(DRDQ(I,J),I=1,NMODES)
1050                CONTINUE
                ENDIF
C
C               Apply the Newton-Raphson scheme to figure the appropriate
C               linear correction in the state vector so as to drive the
C               appropriate residuals to zero. For the steady case, only
C               the steady amplitudes need to be corrected.
C
                CALL SOLVE(DRDQ,RES,DQALL,3*MAXMODE,1,NMODES)
                IF (LNEWT) WRITE(2,'(/A,13(1PE10.2))')
     &          ' DELTA avg amps [m] : ',(-DQALL(I),I=1,NMODES)
C
C               Update the augmented state vector, at the same time
C               checking for convergence of the maximum residual and
C               of the relative change in the state vector QALL.
C
                CONVERGED = .TRUE.
                RESMAX = 0.
                DO 1060 I = 1,NMODES
                    QALL(I) = QALL(I)-DQALL(I)*FACTOR
C
C                   Check relative change in state vector.
C
                    IF (QALL(I).NE.0.) THEN
                        IF (ABS(DQALL(I)/QALL(I)).GT.SMAX)
     &                  CONVERGED=.FALSE.
                    ENDIF
C
```

202

```fortran
C                   Check relative size of residuals.
C
                    IF (ABS(RES(I)/QBIG(I,1)).GT.RMAX) CONVERGED=.FALSE.
                    IF (ABS(RES(I)).GT.ABS(RESMAX)) RESMAX=RES(I)
1060            CONTINUE
C
                IF (LOOPS.GE.ABS(LMAX)) THEN
                    IF (LMAX.LT.0) THEN
1065                    WRITE(*,'(A,$)') ' Continue iterations ? '
                        READ(*,'(A)',ERR=1065) ANSWER
                        CONVERGED=.TRUE.
                        IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) THEN
                            CONVERGED = .FALSE.
                            LOOPS = 0
                        ENDIF
                    ELSE
                        CONVERGED=.TRUE.
                    ENDIF
                ENDIF
C
C           Print current status to screen.
C
1070            FORMAT (A,F6.2,A,I4,A,1PE8.1)
                IF (VLINES) WRITE(*,1070) ' STEADY - AOA =',
     &          DUMMY2,' deg ; Loop',LOOPS,' ; Rmax = ',RESMAX
                IF (.NOT.VLINES) WRITE(*,1070) ' STEADY - VEL =',
     &          DUMMY2,' m/s ; Loop',LOOPS,' ; Rmax = ',RESMAX
C
                IF (LSTART) THEN
                    DO 1075 I = 1,NMODES
                        QLIT(I,2) = 0.
                        QLIT(I,3) = 0.
1075                CONTINUE
                    WRITE(3,*) AOA*180./PI,VEL,0.,((QLIT(I,J),J=1,3),
     &              I=1,NMODES),RESMAX
                ENDIF
            END DO
C
C       Extract the modal amplitudes from the final, converged
C       augmented state vector.
C
        DO 1080 I = 1,NMODES
            QLIT(I,1) = QALL(I)
1080    CONTINUE
        FREQ = 0.
C
C       Calculate the midchord tip deflection components and the tip
C       twist components.
C
        HTIP(1) = 0.
        ATIP(1) = 0.
        DO 1090 I = 1,NB+NT+NC
            HTIP(1) = HTIP(1)+QLIT(I,1)*FMODE(0,'X',I,1.)*
     &      FMODE(0,'Y',I,0.)
            ATIP(1) = ATIP(1)+QLIT(I,1)*FMODE(0,'X',I,1.)*
     &      FMODE(1,'Y',I,0.)/CHORD
1090    CONTINUE
C
        VTIP(1) = 0.
        IF (NF.GT.0) THEN
        DO 1100 I = NB+NT+NC+1,NB+NT+NC+NF
            VTIP(1) = VTIP(1)+QLIT(I,1)*FMODE(0,'X',I,1.)*
     &      FMODE(0,'Y',I,0.)
1100    CONTINUE
```

203

```
          ENDIF
C
C         Convert tip deflection to centimeters and tip twist to degrees.
C
          IF (LATAN) ATIP(1) = ATAN(ATIP(1))
          HTIP(1) = HTIP(1)*100.
          ATIP(1) = ATIP(1)*180./PI
          VTIP(1) = VTIP(1)*100.
C
C         Write converged results.
C
          ANG = AOA*180./PI
          IF (LNEWT) THEN
             WRITE(2,'(/A,I5,A)') ' After',LOOPS,' N-R iterations :'
             DO 1110 I = 1,NMODES
                WRITE(2,*) ' Avg ',MLABEL(I),' amp =',QLIT(I,1),' m'
1110         CONTINUE
             WRITE(2,*) ' H tip =',HTIP(1),' cm'
             WRITE(2,*) ' A tip =',ATIP(1),' deg'
             WRITE(2,*) ' V tip =',VTIP(1),' cm'
             WRITE(2,*) ' AOA =',ANG,' degs'
             WRITE(2,*) ' VEL =',VEL,' m/s'
          ELSE
             WRITE(2,'(2F7.2,2F9.3,$)') VEL,ANG,HTIP(1),ATIP(1)
             IF (NF.GT.0) WRITE(2,'(F9.3,$)') VTIP(1)
             IF (LOOPS.LT.ABS(LMAX)) WRITE(2,'(A)') ' '
             IF (LOOPS.EQ.ABS(LMAX)) WRITE(2,'(A)')
     &          ' * Not converged '
          ENDIF
1999  CONTINUE

C-----FILE: UNSTEADY.INC -----------------------------------------
C
C     Calculate mode number associated with BEN_TOR variable, i.e.
C     either first bending (#1) or first torsion (#NB+1).
C
      MBT = (BEN_TOR-1)*NB + 1
C
C     Set the velocity VEL or the root angle of attack AOA, depending
C     on whether lines of constant velocity or constant angle.
C
      IF (VLINES) VEL = DUMMY1
      IF (.NOT.VLINES) AOA = DUMMY1*PI/180.
C
C
C     If unsteady, read in the start & end values and the incremental
C     step size of the amplitude of oscillating twist for each
C     line of constant velocity/root angle.
C
      IF ((DUMMY1.EQ.DUM1LO).OR.(.NOT.LFLUTB)) THEN
2010     IF (BEN_TOR.EQ.1) THEN
             IF (VLINES) WRITE(*,'(A,F5.1,A,$)') ' VEL =',DUMMY1,
     &          ' ; Bending amplitude start, end, & step size (cm) ? '
             IF (.NOT.VLINES) WRITE(*,'(A,F5.1,A,$)') ' AOA =',
     &          DUMMY1,' ; Bending amplitude start, end, & step '//
     &          'size (cm) ? '
         ELSE
             IF (VLINES) WRITE(*,'(A,F5.1,A,$)') ' VEL =',DUMMY1,
     &          ' ; Twist amplitude start, end, & step size (deg) ? '
             IF (.NOT.VLINES) WRITE(*,'(A,F5.1,A,$)') ' AOA =',
     &          DUMMY1,' ; Twist amplitude start, end, & step '//
     &          'size (deg) ? '
         ENDIF
         READ(*,*,ERR=2010) DUM2LO,DUM2HI,DUM2INC
```

204

```
            IF ((.NOT.VLINES).AND.(DUM2LO.LT.0.)) GOTO 2010
            IF ((.NOT.VLINES).AND.(DUM2HI.LT.0.)) GOTO 2010
            IF ((DUM2HI-DUM2LO)/DUM2INC.LT.0.) GOTO 2010
        ENDIF
C
C       Determine if previous values should be used as an initial guess.
C
        IF (DUMMY1.EQ.DUM1LO) THEN
            ANSWER = 'N'
        ELSEIF (LFLUTB) THEN
            ANSWER = 'Y'
        ELSEIF (.NOT.LFLUTB) THEN
2020        WRITE(*,'(A,$)') ' Use previous values as '//
     &        'initial guess ? '
            READ(*,'(A1)',ERR=2020) ANSWER
        ENDIF
C
        IF ((ANSWER.NE.'Y').AND.(ANSWER.NE.'y')) THEN
C
C       Read in the initial guess for root angle AOA, or for
C       velocity VEL — to be used for the first iteration of the
C       Newton-Raphson solver for the first corresponding line
C       of constant velocity/root angle — and insert in location
C       reserved for sine harmonic of bending/twist amplitude.
C
2021        WRITE(*,'(A,$)') ' Start from restart file ? '
            READ(*,'(A)',ERR=2021) ANSWER
C
            QLIT(1,1) = 0.
            IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) THEN
2022            WRITE(*,'(A,$)') ' Restart filename ? '
                READ(*,'(A)',ERR=2022) FILENAME
                OPEN(UNIT=4,FILE=FILENAME,STATUS='OLD',
     &            FORM='FORMATTED',ERR=2022)
2023            READ(4,*,ERR=2022,END=2029) AOADUM,VELDUM,FRQDUM,
     &            ((QDUM(I,J),J=1,3),I=1,NMODES)
                IF (((VLINES).AND.(DUMMY1.EQ.VELDUM)).OR.
     &            ((.NOT.VLINES).AND.(DUMMY1.EQ.AOADUM))) THEN
                    DO 2025 I = 1,NMODES
                    DO 2025 J = 1,3
                        QLIT(I,J) = QDUM(I,J)
                        QALL((J-1)*NMODES+I) = QLIT(I,J)
2025                CONTINUE
                    IF (VLINES) THEN
                        AOA = AOADUM*PI/180.
                        QALL(NMODES+MBT) = AOA
                    ELSE
                        VEL = VELDUM
                        QALL(NMODES+MBT) = VEL**2
                    ENDIF
2026                IF (FRQDUM.LE.0.) THEN
                        WRITE(*,'(A,$)') ' Initial frequency guess '//
     &                    '(in Hz) ? '
                        READ(*,*,ERR=2026) FRQDUM
                        GOTO 2026
                    ENDIF
                    FREQ = FRQDUM*2.*PI*(CHORD/2.)/VEL
                    QALL(2*NMODES+MBT) = FREQ
                ENDIF
                GOTO 2023
2029            CLOSE(4)
                IF (QLIT(1,1).EQ.0.) GOTO 2021
            ELSE
C
```

```
              IF (VLINES) THEN
2030            WRITE(*,'(A,$)') ' Initial root angle guess (deg) ? '
                READ(*,*,ERR=2030) AOA
                AOA = AOA*PI/180.
                QALL(NMODES+MBT) = AOA
              ELSE
2040            WRITE(*,'(A,$)') ' Initial velocity guess (m/s) ? '
                READ(*,*,ERR=2040) VEL
                IF (VEL.LT.0.) GOTO 2040
                QALL(NMODES+MBT) = VEL**2
              ENDIF
C
C         Query user if he wants to directly input the initial average
C         amplitudes (eg. might be needed if flutter is near the
C         divergence velocity).
C
              WRITE(*,'(A,$)') ' Input initial average deflections ? '
              READ(*,'(A1)') ANSWER
              DO 2060 I = 1,NMODES
                IF (DUMMY1.EQ.DUM1LO) QLIT(I,1) = 0.
                IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) THEN
2050              WRITE(*,'(A,$)') '    Mode '//MLABEL(I)//
     &                ' average [m] = '
                  READ(*,*,ERR=2050) QLIT(I,1)
                ENDIF
                QALL(I) = QLIT(I,1)
2060          CONTINUE
C
2070          WRITE(*,'(A,$)') ' Start vibration on eigenmode ? '
              READ(*,'(A)',ERR=2070) ANSWER
              IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) THEN
C
C             Determine current nonlinear mass & stiffness matrices.
C
                DO 2073 I1 = 1,MAXMODE
                DO 2073 I2 = I1,MAXMODE
                  MDUM(I1,I2) = M(I1,I2)
                  KDUM(I1,I2) = K(I1,I2)
C
C                 Add corrections to stiffness matrix for geometric
C                 nonlinearities.
C
                  IF (LGEOM) THEN
                    IF ((I1.GT.NB).AND.(I1.LE.NB+NT).AND.
     &                 (I2.GT.NB).AND.(I2.LE.NB+NT)) THEN
                       DO 2071 I3 = 1,NB
                       DO 2071 I4 = 1,NB
                         KDUM(I1,I2) = KDUM(I1,I2) +
     &                     R(I3,I4,I1-NB,I2-NB)*QLIT(I3,1)*QLIT(I4,1)
2071                   CONTINUE
                    ELSEIF ((I1.GT.NB).AND.(I1.LE.NB+NT).AND.
     &                 (I2.GT.NB+NT+NC).AND.(I2.LE.NB+NT+NC+NF)) THEN
                       DO 2072 I3 = 1,NB
                         KDUM(I1,I2) = KDUM(I1,I2) +
     &                     H(I3,I1-NB,I2-NB-NT-NC)*QLIT(I3,1)
2072                   CONTINUE
                    ENDIF
                  ENDIF
C
                  MDUM(I2,I1) = MDUM(I1,I2)
                  KDUM(I2,I1) = KDUM(I1,I2)
2073            CONTINUE
C
C               Add contribution to stiffness matrix for cubic stiffening.
```

206

```
c
                KDUM(NB+1,NB+1) = KDUM(NB+1,NB+1) + KTTCUBE*
       &           (3.*QLIT(NB+1,1)**2+.75*QLIT(NB+1,2)**2+
       &           .75*QLIT(NB+1,3)**2)
c
c               Call EISPACK eigenvalue solver.
c
                CALL RSG(MAXMODE,NMODES,KDUM,MDUM,FVIB,1,QVIB,
       &          FV1,FV2,IERR)
c
c               Print out choice of eigenmodes to start from (negative
c               frequency to indicate imaginary).
c
                DO 2076 I = 1,NMODES
                   QMAX = QVIB(1,I)
                   DO 2075 J = 1,NMODES
                      IF (ABS(QVIB(J,I)).GT.ABS(QMAX))
       &                 QMAX = QVIB(J,I)
2075               CONTINUE
                   WRITE(*,'(5X,I2,A,F9.2,A,SP,32(1X,F4.1))')
       &             I,' Freq =',SQRT(ABS(FVIB(I)))/(2.*PI)*
       &             (FVIB(I)/ABS(FVIB(I))),' Hz ; mode shape = ',
       &             (QVIB(J,I)/QMAX,J=1,NMODES)
2076            CONTINUE
c
2077            WRITE(*,'(/A,$)') ' Starting eigenmode number ? '
                READ(*,*,ERR=2077) IEIG
                IF ((IEIG.LE.0).OR.(IEIG.GT.NMODES)) GOTO 2077
c
c               Initialize amplitudes of vibration.
c
                TWIST = DUM2LO*PI/180.
                DEFLC = DUM2LO/100.
                DO 2078 I = 1,NMODES
                   IF (BEN_TOR.EQ.1) QLIT(I,2) = DEFLC/
       &             FMODE(0,'X',MBT,1.)/FMODE(0,'Y',MBT,0.)*
       &             QVIB(I,IEIG)/QVIB(MBT,IEIG)
                   IF (BEN_TOR.EQ.2) QLIT(I,2) = TAN(TWIST)*CHORD/
       &             FMODE(0,'X',MBT,1.)/FMODE(1,'Y',MBT,0.)*
       &             QVIB(I,IEIG)/QVIB(MBT,IEIG)
                   IF (I.NE.MBT) QALL(NMODES+I) = QLIT(I,2)
                   QLIT(I,3) = 0.
                   IF (I.NE.MBT) QALL(2*NMODES+I) = QLIT(I,3)
2078            CONTINUE
c
c               Initialize reduced frequency.
c
                FREQ = SQRT(FVIB(IEIG))*(CHORD/2.)/VEL
                QALL(2*NMODES+MBT) = FREQ
             ELSE
c
c               Initialize reduced frequency.
c
                IF (DUMMY1.NE.DUM1LO) WRITE(*,*) ' Current frequency =',
       &          FREQ*VEL/(2.*PI*(CHORD/2.)),' Hz'
2080            WRITE(*,'(A,$)') ' Initial frequency guess (in Hz) ? '
                READ(*,*,ERR=2080) FREQ
                IF (FREQ.LE.0.) GOTO 2080
                FREQ = FREQ*2.*PI*(CHORD/2.)/VEL
                QALL(2*NMODES+MBT) = FREQ
c
c               Initialize the oscillating amplitudes to zero.
c
                DO 2085 J = 2,3
```

```
                  DO 2085 I = 1,NMODES
                      QLIT(I,J) = 0.
                      IF (I.NE.MBT) QALL(NMODES*(J-1)+I) = QLIT(I,J)
      2085        CONTINUE
                ENDIF
              ENDIF
           ENDIF
      C
      C     Read in the non-dimensional step size tolerance [maximum
      C     delta(X)/X] to be applied to the root angle/velocity and
      C     frequency corrections in relaxing the Newton-Raphson solver.
      C
            IF ((DUMMY1.EQ.DUM1LO).OR.(.NOT.LFLUTB)) THEN
      2090      WRITE(*,'(A,$)') ' Step size tolerance ? '
                READ(*,*,ERR=2090) TOL
                IF (TOL.LE.0.) GOTO 2090
            ENDIF
      C
            DO 2999 JGEOM = 0,IGEOM
            RGEOM = 1.
            IF ((LGEOM).AND.(IGEOM.GT.0)) THEN
                RGEOM = REAL(JGEOM)/REAL(IGEOM)
                WRITE(*,'(A,F4.3)') 'RGEOM = ',RGEOM
            ENDIF
      C
      C     Loop through the appropriate variable, denoted by the dummy
      C     variable DUMMY2, for each line of constant velocity/root angle.
      C
            DO 2999 DUMMY2 = DUM2LO,DUM2HI,DUM2INC
      C
      C        Initialize the number of iterations to zero.
      C
               LOOPS = 0
      C
      C        Initialize convergence. If zero velocity, automatically set
      C        all amplitudes to zero and skip Newton-Raphson solver.
      C
               CONVERGED = .FALSE.
               IF ((.NOT.VLINES).AND.(DUMMY2.EQ.0.)) THEN
                   DO 2100 I = 1,NMODES
                       QLIT(I,1) = 0.
                       QALL(I) = 0.
      2100         CONTINUE
                   CONVERGED = .TRUE.
               ENDIF
      C
      C        Rescale unsteady, variable amplitudes from previous values
      C        according to new set amplitude.
      C
               DO 2110 I = 1,NMODES
               DO 2110 J = 2,3
                   IF ((I.NE.MBT).AND.(DUMMY2.NE.DUM2LO)) QALL(NMODES*
           &         (J-1)+I)=QALL(NMODES*(J-1)+I)*DUMMY2/(DUMMY2-DUM2INC)
      2110     CONTINUE
      C
      C        Loop through the Newton-Raphson scheme until it is
      C        converged to an acceptable limit.
      C
               DO WHILE (.NOT.CONVERGED)
      C
      C           Extract the modal amplitudes from
      C           the augmented modal amplitude vector.
      C
                  DO 2120 I = 1,NMODES
```

208

```
              DO 2120 J = 1,3
                  QLIT(I,J) = QALL(NMODES*(J-1)+I)
2120          CONTINUE
C
C             Extract current value of unknown root angle/velocity
C             from the augmented state vector QALL, appropriate to lines
C             of constant velocity or root angle. Set velocity to
C             zero if Newton-Raphson solver drives VEL**2 below zero.
C
              IF (VLINES) THEN
                  AOA = QALL(NMODES+MBT)
              ELSEIF (QALL(NMODES+MBT).GT.0.) THEN
                  VEL = SQRT(QALL(NMODES+MBT))
              ELSE
                  QALL(NMODES+MBT) = 0.
                  VEL = 0.
              ENDIF
C
C             Extract current value of the unknown reduced frequency
C             from the augmented state vector QALL.
C
              FREQ = QALL(2*NMODES+MBT)
C
C             Extract the desired twist oscillating amplitudes from
C             the dummy variable DUMMY2.
C
              TWIST = DUMMY2*PI/180.
              DEFLC = DUMMY2/100.
              IF (BEN_TOR.EQ.1) QLIT(MBT,2) = DEFLC/
     &         FMODE(0,'X',MBT,1.)/FMODE(0,'Y',MBT,0.)
              IF (BEN_TOR.EQ.2) QLIT(MBT,2) = TAN(TWIST)*CHORD/
     &         FMODE(0,'X',MBT,1.)/FMODE(1,'Y',MBT,0.)
              QLIT(MBT,3) = 0.
C
              LOOPS = LOOPS+1
C
C             Write current values of inputs to residual calculations.
C
              IF (LNEWT) THEN
                  WRITE(2,'(/A,I4,11X,13(4X,A2,4X))') ' LOOP = ',
     &             LOOPS,(MLABEL(I),I=1,NMODES)
                  DO 2130 J = 1,3
                      IF (J.EQ.1) WRITE(2,'(A,$)') ' Avg '
                      IF (J.EQ.2) WRITE(2,'(A,$)') ' Sin '
                      IF (J.EQ.3) WRITE(2,'(A,$)') ' Cos '
                      WRITE(2,'(A,13(1PE10.2))') 'modal amp : ',
     &                 (QLIT(I,J),I=1,NMODES)
2130              CONTINUE
                  WRITE(2,*) ' VEL =',VEL,' m/s'
                  WRITE(2,*) ' AOA =',AOA*180./PI,' degs'
                  WRITE(2,*) '   k =',FREQ
                  OMEGA = FREQ*VEL/(CHORD/2.)/(2.*PI)
                  WRITE(2,*) '   w =',OMEGA,' Hz'
              ENDIF
C
C             Calculate the residuals from subroutine RESIDUAL, which
C             are functions of the velocity VEL, root angle of attack AOA,
C             reduced frequency FREQ, and modal amplitudes QLIT.
C
              CALL RESIDUAL(VEL,AOA,FREQ,QLIT,RGEOM,RES,QBIG)
C
C             Write current values of residuals.
C
              IF (LNEWT) THEN
```

```
                   DO 2140 J = 1,3
                      IF (J.EQ.1) WRITE(2,'(/A,$)') ' Avg '
                      IF (J.EQ.2) WRITE(2,'(A,$)') ' Sin '
                      IF (J.EQ.3) WRITE(2,'(A,$)') ' Cos '
                      WRITE(2,'(A,13(1PE10.2))') 'modal Qs   : ',
          &              (QBIG(I,J),I=1,NMODES)
     2140         CONTINUE
                   DO 2145 J = 1,3
                      IF (J.EQ.1) WRITE(2,'(/A,$)') ' Avg '
                      IF (J.EQ.2) WRITE(2,'(A,$)') ' Sin '
                      IF (J.EQ.3) WRITE(2,'(A,$)') ' Cos '
                      WRITE(2,'(A,13(1PE10.2))') 'residuals : ',
          &              (RES(I),I=(J-1)*NMODES+1,J*NMODES)
     2145         CONTINUE
                ENDIF
     C
     C
     C          Calculate the derivative matrix of the residuals wrt the
     C          modal amplitudes using subroutine R_REDIV, which is a
     C          function of the velocity VEL, root angle of attack AOA,
     C          reduced frequency FREQ, and modal amplitudes QLIT. The
     C          current values of the residuals RES are also passed since
     C          the derivative matrix may be calculated numerically, in
     C          which case the current values are needed.
     C
                CALL R_DERIV(BEN_TOR,VEL,AOA,FREQ,QLIT,RES,?GEOM,DRDQ)
     C
     C          Write derivative matrix.
     C
                IF (LNEWT) THEN
                   WRITE(2,'(/A)') ' NUMERIC dR/dq MATRIX :'
                   WRITE(2,'(12X,99(6X,I1,3X))')
          &          ((I,I=1,NMODES),J=1,3)
                   DO 2160 I1 = 1,3
                   DO 2160 I2 = 1,NMODES
                      IF (I1.EQ.1) CDUM = 'dR/dq'//MLABEL(I2)//'o'
                      IF (I1.EQ.2) CDUM = 'dR/dq'//MLABEL(I2)//'s'
                      IF (I1.EQ.3) CDUM = 'dR/dq'//MLABEL(I2)//'c'
                      IF ((I1.EQ.2).AND.(I2.EQ.MBT)) THEN
                         IF (VLINES) CDUM = ' dR/dAOA'
                         IF (.NOT.VLINES) CDUM = ' dR/dV*2'
                      ELSEIF ((I1.EQ.3).AND.(I2.EQ.MBT)) THEN
                         CDUM = '    dR/dk'
                      ENDIF
                      J = (I1-1)*NMODES+I2
                      WRITE(2,'(4X,A8,13(1PE10.2))')
          &            CDUM,(DRDQ(I,J),I=1,3*NMODES)
     2160         CONTINUE
                ENDIF
     C
     C          Apply the Newton-Raphson scheme to figure the appropriate
     C          linear correction in the state vector so as to drive the
     C          appropriate residuals to zero. For the steady case, only
     C          the steady amplitudes need to be corrected.
     C
                CALL SOLVE(DRDQ,RES,DQALL,3*MAXMODE,1,3*NMODES)
     C
     C          Write the uncorrected state vector corrections.
     C
                IF (LNEWT) THEN
                   DO 2165 I = 1,3
                      IF (I.NE.1) RDUM = DQALL((I-1)*NMODES+MBT)
                      IF (I.NE.1) DQALL((I-1)*NMODES+MBT) = 0.
                      IF (I.EQ.1) WRITE(2,'(/A,$)') ' DELTA avg '
                      IF (I.EQ.2) WRITE(2,'(A,$)')  ' DELTA sin '
```

210

```fortran
                        IF (I.EQ.3) WRITE(2,'(A,$)')   ' DELTA cos '
                        WRITE(2,'(A,13(1PE10.2))') 'amps [m] : ',
     &                    (-DQALL(J),J=(I-1)*NMODES+1,I*NMODES)
                        IF (I.NE.1) DQALL((I-1)*NMODES+MBT) = RDUM
 2165            CONTINUE
                IF (VLINES) WRITE(2,*) ' DELTA AOA =',
     &            -DQALL(NMODES+MBT),' deg'
                IF (.NOT.VLINES) WRITE(2,*) ' DELTA V*2 =',
     &            -DQALL(NMODES+MBT),' (m/s)**2'
                WRITE(2,*) ' DELTA k   =',-DQALL(2*NMODES+MBT)
              ENDIF
C
C               Calculate the appropriate factor for relaxation when the
C               correction step size is too large for either the root
C               angle/velocity or reduced frequency.
C
                FACTOR = 1.
                DO 2170 I = 1,2
                  J = I*NMODES + MBT
                  IF (QALL(J).NE.0.) THEN
                    IF (ABS(DQALL(J)/FACTOR/QALL(J)).GT.TOL)
     &                FACTOR = ABS(DQALL(J)/(TOL*QALL(J)))
                  ENDIF
 2170           CONTINUE
                IF (LNEWT) WRITE(2,'(/A,1PE10.2)')
     &            ' FACTOR = ',FACTOR
C
C               Update the augmented state vector, at the same time
C               checking for convergence of the maximum residual and
C               of the relative change in the state vector QALL.
C
                CONVERGED = .TRUE.
                RESMAX = 0.
                DO 2180 I = 1,NMODES
                DO 2180 J = 1,3
                  II = (J-1)*NMODES+I
                  QALL(II) = QALL(II)-DQALL(II)/FACTOR
C
C                 Check relative change in state vector.
C
                  IF (QALL(II).NE.0.) THEN
                    IF (ABS(DQALL(II)/QALL(II)).GT.SMAX)
     &                CONVERGED=.FALSE.
                  ENDIF
C
C                 Check relative size of residuals.
C
                  IF (ABS(RES(II)/QBIG(I,J)).GT.RMAX) CONVERGED=.FALSE.
                  IF (ABS(RES(II)).GT.ABS(RESMAX)) RESMAX=RES(II)
 2180           CONTINUE
C
C               Print current status to screen.
C
                IF (LFLUTB) THEN
                  IF (VLINES) WRITE(*,'(A,F6.2,$)')
     &              ' VEL =',DUMMY1
                  IF (.NOT.VLINES) WRITE(*,'(A,F6.2,$)')
     &              ' AOA =',DUMMY1
                ELSE
                  WRITE(*,'(A,F6.2,$)') ' AMP =',DUMMY2
                ENDIF
 2190           FORMAT (A,I4,A,1PE8.1,A,0PF6.2,A,F5.2,A,F4.2,A)
                IF (VLINES) THEN
                  WRITE(*,2190) ' ; Loop ',LOOPS,' ; Rmax = ',
```

```
    &                    RESMAX,' ; AOA =',QALL(NMODES+MBT)*180./PI,
    &                    ' deg ; w =',QALL(2*NMODES+MBT)*VEL/(CHORD/
    &                    2.)/(2.*PI),' Hz (k=',QALL(2*NMODES+MBT),')'
                    ELSE
                        WRITE(*,2190) ' ; Loop ',LOOPS,' ; Rmax = ',
    &                    RESMAX,' ; VEL =',SQRT(QALL(NMODES+MBT)),
    &                    ' m/s; w =',QALL(2*NMODES+MBT)*VEL/(CHORD/
    &                    2.)/(2.*PI),' Hz (k=',QALL(2*NMODES+MBT),')'
                    ENDIF
C
                    OMEGA = FREQ*VEL/(CHORD/2.)/(2.*PI)
                    IF (LSTART) WRITE(3,*) AOA*180./PI,VEL,OMEGA,
    &                ((QLIT(I,J),J=1,3),I=1,NMODES),RESMAX
C
                    IF (LOOPS.GE.ABS(LMAX)) THEN
                        IF (LMAX.LT.0) THEN
2195                        WRITE(*,'(A,$)') ' Continue iterations ? '
                            READ(*,'(A)',ERR=2195) ANSWER
                            CONVERGED=.TRUE.
                            IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) THEN
                                CONVERGED = .FALSE.
                                LOOPS = 0
2196                            WRITE(*,'(A,F5.4,A,$)') ' TOL = ',TOL,
    &                            ' ; New step size tolerance ? '
                                READ(*,*,ERR=2196) TOL
                                IF (TOL.LE.0.) GOTO 2196
                            ENDIF
                        ELSE
                            CONVERGED=.TRUE.
                        ENDIF
                    ENDIF
                END DO
C
C           Extract the modal amplitudes and the velocity and reduced
C           frequency from the final, converged augmented state vector.
C
                DO 2200 I = 1,NMODES
                DO 2200 J = 1,3
                    QLIT(I,J) = QALL(NMODES*(J-1)+I)
2200            CONTINUE
C
C           Extract the appropriate root angle/velocity,
C           frequency, and twist oscillating amplitudes from the
C           final, converged augmented state vector.
C
                QLIT(MBT,3) = 0.
                IF (BEN_TOR.EQ.1) QLIT(MBT,2) = DEFLC/
    &            FMODE(0,'X',MBT,1.)/FMODE(0,'Y',MBT,0.)
                IF (BEN_TOR.EQ.2) QLIT(MBT,2) = TAN(TWIST)*CHORD/
    &            FMODE(0,'X',MBT,1.)/FMODE(1,'Y',MBT,0.)
                IF (VLINES) AOA = QALL(NMODES+MBT)
                IF (.NOT.VLINES) VEL = SQRT(QALL(NMODES+MBT))
                FREQ = QALL(2*NMODES+MBT)
C
C           Calculate the midchord tip deflection components and the tip
C           twist components.
C
                DO 2220 J = 1,3
                    HTIP(J) = 0.
                    ATIP(J) = 0.
                    DO 2210 I = 1,NB+NT+NC
                        HTIP(J) = HTIP(J)+QLIT(I,J)*FMODE(0,'X',I,1.)*
    &                    FMODE(0,'Y',I,0.)
                        ATIP(J) = ATIP(J)+QLIT(I,J)*FMODE(0,'X',I,1.)*
```

212

```
     &                    FMODE(1,'Y',I,0.)/CHORD
2210          CONTINUE
C
              VTIP(J) = 0.
              IF (NF.GT.0) THEN
              DO 2215 I = NB+NT+NC+1,NB+NT+NC+NF
                  VTIP(J) = VTIP(J)+QLIT(I,J)*FMODE(0,'X',I,1.)*
     &                FMODE(0,'Y',I,0.)
2215          CONTINUE
              ENDIF
C
C        Convert tip deflection to centimeters and tip twist to
C        degrees.
C
              IF (LATAN) ATIP(J) = ATAN(ATIP(J))
              HTIP(J) = HTIP(J)*100.
              ATIP(J) = ATIP(J)*180./PI
              VTIP(J) = VTIP(J)*100.
2220     CONTINUE
C
C        Write final/converged results.
C
         ANG = AOA*180./PI
         OMEGA = FREQ*VEL/(CHORD/2.)/(2.*PI)
         IF ((.NOT.LNEWT).AND.(BEN_TOR.EQ.1)) THEN
             WRITE(2,'(3F7.2,5F9.3,$)') VEL,ANG,OMEGA,
     &         (HTIP(J),J=1,2),ATIP(1),SQRT(ATIP(2)**2+ATIP(3)**2),
     &         ATAN2(ATIP(2),ATIP(3))
             IF (NF.GT.0) WRITE(2,'(3F9.3,$)') VTIP(1),
     &         SQRT(VTIP(2)**2+VTIP(3)**2),ATAN2(VTIP(2),VTIP(3))
             IF (LOOPS.LT.ABS(LMAX)) WRITE(2,'(A)') ' '
             IF (LOOPS.GE.ABS(LMAX)) WRITE(2,'(A)')
     &         ' * Not converged'
         ELSEIF ((.NOT.LNEWT).AND.(BEN_TOR.EQ.2)) THEN
             WRITE(2,'(3F7.2,5F9.3,$)') VEL,ANG,OMEGA,
     &         HTIP(1),SQRT(HTIP(2)**2+HTIP(3)**2),
     &         ATAN2(HTIP(2),HTIP(3)),ATIP(1),ATIP(2)
             IF (NF.GT.0) WRITE(2,'(3F9.3,$)') VTIP(1),
     &         SQRT(VTIP(2)**2+VTIP(3)**2),ATAN2(VTIP(2),VTIP(3))
             IF (LOOPS.LT.ABS(LMAX)) WRITE(2,'(A)') ' '
             IF (LOOPS.GE.ABS(LMAX)) WRITE(2,'(A)')
     &         ' * Not converged'
         ELSE
             WRITE(2,'(/A,I5,A)') ' After',LOOPS,' N-R iterations :'
             WRITE(2,'(16X,13(4X,A2,4X))') (MLABEL(I),I=1,NMODES)
             WRITE(2,'(A,13(1PE10.2))') ' Avg amps [m] : ',
     &         (QLIT(I,1),I=1,NMODES)
             WRITE(2,'(A,13(1PE10.2))') ' Sin amps [m] : ',
     &         (QLIT(I,2),I=1,NMODES)
             WRITE(2,'(A,13(1PE10.2))') ' Cos amps [m] : ',
     &         (QLIT(I,3),I=1,NMODES)
             WRITE(2,'(/A,F7.3,A)') ' AOA =',ANG,' degs'
             WRITE(2,'(A,F7.3,A)')  ' VEL =',VEL,' m/s'
             WRITE(2,'(A,F7.3)')    '   k =',FREQ
             WRITE(2,'(A,F6.3,A)')  '   w = ',OMEGA,' Hz'
         ENDIF
2999  CONTINUE

C-----FILE : V_G.FOR --------------------------------------------------
C
      PROGRAM VG_ANALYSIS
C
      INCLUDE     PARAM.INC
      INCLUDE     GLBBLK.INC
```

```
      REAL        TRATIO,FREQ,KDET
      REAL        KINV(MAXMODE,MAXMODE),FV3(MAXMODE)
      REAL        AR(MAXMODE,MAXMODE),AI(MAXMODE,MAXMODE)
      REAL        ZR(MAXMODE),ZI(MAXMODE),QB(NBMAX)
      REAL        QR(MAXMODE,MAXMODE),QI(MAXMODE,MAXMODE)
      REAL        OMEGA(MAXMODE),DAMP(MAXMODE),VEL(MAXMODE)
      REAL        SL,KVL,LAM,SIGL,ALFA,SM,KVM,SIGM
      COMPLEX     IC,L1L2,L3L4,M1M2,M3M4,THEO
      COMPLEX     A(MAXMODE,MAXMODE)
      COMPLEX     KINVA(MAXMODE,MAXMODE)
      INTEGER     IERR,IP(MAXMODE)
      LOGICAL     LSTRUC,CONVERGED
      CHARACTER   LAYUP*25,ANSWER*1,FILENAME*25
C
      FOIL = 'NAC12'
      IC = (0.,1.)
      ATYPE = 1
      LINEAR = .TRUE.
      STEADY = .FALSE.
      REDUC = .TRUE.
      CORREC = .TRUE.
      VLINES = .TRUE.
      LATAN = .FALSE.
      LAEROF = .FALSE.
      LCONST = .TRUE.
      LGEOM = .TRUE.
C
C     Read in the layup.
C
10    WRITE(*,'(/A,$)') 'Layup : '
      READ(*,'(A)',ERR=10) LAYUP
C
C     Read in number of mode shapes to be used for the analysis.
C
20    WRITE(*,'(A,$)') 'Number of out-of-plane, '//
     & 'torsion, & fore-&-aft modes ? '
      READ(*,*,ERR=20) NB,NT,NF
      NC = 0
      IF ((NB.LT.1).OR.(NB.GT.NBMAX)) GOTO 20
      IF ((NT.LT.1).OR.(NT.GT.NTMAX)) GOTO 20
      IF ((NF.LT.0).OR.(NF.GT.NFMAX)) GOTO 20
      NMODES = NB+NT+NC+NF
C
      LSTRUC = .FALSE.
30    WRITE(*,'(A,$)') 'Output structures troubleshooting '//
     & 'diagnostics ? '
      READ(*,'(A)',ERR=30) ANSWER
      IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) LSTRUC = .TRUE.
C
C     Create mass and stiffness matrices by calling STATIC subroutine.
C
      CALL STATIC(LAYUP,LSTRUC,TRATIO,IERR)
      IF (IERR.NE.0) THEN
         WRITE(*,'(A,I2,A)') ' IOSTAT=',IERR,' error reading '//
     &   TRIM(LAYUP)//'.DAT data file.'
         GOTO 10
      ENDIF
C
C     Open output file.
C
      FILENAME = TRIM(LAYUP)//'WNAV.VG'
      OPEN(UNIT=2,FILE=TRIM(FILENAME),STATUS='NEW',
     & FORM='FORMATTED',IOSTAT=IERR)
      IF (IERR.EQ.0) THEN
```

214

```fortran
              WRITE(*,*) 'Analysis results being sent to '//TRIM(FILENAME)
          ELSE
              WRITE(*,'(A,I2,A)') 'IOSTAT=',IERR,' error opening '//
     &        TRIM(FILENAME)//' as output file.'
              GOTO 10
          ENDIF
          WRITE(2,'(3I2,A)') NB,NT,NF,' = out-of-plane, torsion, '//
     &    'fore-&-aft modes'
          WRITE(2,'(F5.2,A)') TRATIO,' = NACA airfoil thickness ratio'
C
C     Add geometric nonlinearities to stiffness matrix.
C
          DO 50 I = 1,NB
40            WRITE(*,'(A,I1,A,$)') '    q',I,'B [m] = '
              READ(*,*,ERR=40) QB(I)
50        CONTINUE
C
          DO 80 I1 = 1,MAXMODE
          DO 80 I2 = I1,MAXMODE
              IF ((I1.GT.NB).AND.(I1.LE.NB+NT).AND.
     &        (I2.GT.NB).AND.(I2.LE.NB+NT)) THEN
                  DO 60 I3 = 1,NB
                  DO 60 I4 = 1,NB
                      K(I1,I2) = K(I1,I2) + R(I3,I4,I1-NB,I2-NB)*
     &                QB(I3)*QB(I4)
60                CONTINUE
              ELSEIF ((I1.GT.NB).AND.(I1.LE.NB+NT).AND.
     &        (I2.GT.NB+NT).AND.(I2.LE.NB+NT+NF)) THEN
                  DO 70 I3 = 1,NB
                      K(I1,I2) = K(I1,I2) + H(I3,I1-NB,
     &                I2-NB-NT)*QB(I3)
70                CONTINUE
              ENDIF
              K(I2,I1) = K(I1,I2)
80        CONTINUE
C
C     Loop through reduced frequencies from 0 to 2.
C
          CALL COEFS_LIN('M',0.,SM,KVM,LAM,SIGM,ALFA)
          CALL COEFS_LIN('L',0.,SL,KVL,LAM,SIGL,ALFA)
          DO 999 I = 0,210
C
              FREQ = REAL(I-9)/100.
              IF (I.GE.159) FREQ = 1.5*2.**(I-159)
              IF (I.LE.10) FREQ = REAL(I)/1000.
              IF (I.EQ.0) FREQ = 1.D-4
              WRITE(*,'(A,I3,A,1PG9.3)') 'I = ',I,' ;  k = ',FREQ
C
              THEO = (LAM+IC*(ALFA*FREQ))/(LAM+IC*FREQ)
              L1L2 = 1. - IC/FREQ*SLOPE('L')/SL*THEO
              L3L4 = .5 - KVL/SL + THEO/FREQ**2*SLOPE('L')/SL +
     &        IC/FREQ*(1.+(SIGL/SL-SLOPE('L')/2./SL)*THEO)
C
              M1M2 = .5 + 2.*SM/SL - IC/FREQ*SLOPE('L')/2./SL*THEO
              M3M4 = .25 - KVL/2./SL + SM/SL - 2.*KVM/SL +
     &        THEO/FREQ**2*SLOPE('L')/2./SL + IC/FREQ*(.5 +
     &        2.*SM/SL + 2.*SIGM/SL + .5*(SIGL/SL-SLOPE('L')/
     &        2./SL)*THEO)
C
              IF (REDUC) THEN
                  L1L2 = L1L2/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
                  L3L4 = L3L4/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
                  M1M2 = M1M2/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
                  M3M4 = M3M4/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
```

```
              ENDIF
C
C             Out-of-plane/out-of-plane matrix components.
C
              DO 90 II = 1,NB
              DO 90 JJ = II,NB
                 A(II,JJ) = M(II,JJ) + SL*RHOA*LENGTH*
     &            (CHORD/2.)**2*L1L2*SC_INT(II,JJ)
                 A(JJ,II) = A(II,JJ)
90            CONTINUE
C
C             Out-of-plane/torsion matrix components.
C
              DO 100 II = 1,NB
              DO 100 JJ = NB+1,NB+NT
                 A(II,JJ) = M(II,JJ) + SL*RHOA*LENGTH*
     &            (CHORD/2.)**3/CHORD*L3L4*SC_INT(II,JJ)
100           CONTINUE
C
C             Torsion/out-of-plane matrix components.
C
              DO 110 II = NB+1,NB+NT
              DO 110 JJ = 1,NB
                 A(II,JJ) = M(II,JJ) + SL*RHOA*LENGTH*
     &            (CHORD/2.)**3/CHORD*M1M2*SC_INT(II,JJ)
110           CONTINUE
C
C             Torsion/torsion matrix components.
C
              DO 120 II = NB+1,NB+NT
              DO 120 JJ = II,NB+NT
                 A(II,JJ) = M(II,JJ) + SL*RHOA*LENGTH*
     &            (CHORD/2.)**4/CHORD**2*M3M4*
     &             SC_INT(II,JJ)
                 A(JJ,II) = A(II,JJ)
120           CONTINUE
C
C             Fore-&-aft matrix components
C
              IF (NF.GT.0) THEN
              DO 130 II = 1,NB+NT+NF
              DO 130 JJ = MAX0(II,NB+NT+1),NB+NT+NF
                 A(II,JJ) = M(II,JJ)
                 A(JJ,II) = A(II,JJ)
130           CONTINUE
              ENDIF
C
              CALL INVERT(K,KINV,MAXMODE,NMODES)
C
              DO 150 II = 1,NMODES
              DO 150 KK = 1,NMODES
                 KINVA(II,KK) = (0.,0.)
                 DO 140 JJ = 1,NMODES
                    KINVA(II,KK) = KINVA(II,KK) + KINV(II,JJ)*A(JJ,KK)
140              CONTINUE
                 AR(II,KK) = REAL(KINVA(II,KK))
                 AI(II,KK) = AIMAG(KINVA(II,KK))
150           CONTINUE
C
              CALL CG(MAXMODE,NMODES,AR,AI,ZR,ZI,0,QR,QI,FV1,FV2,FV3,IERR)
C
C             Extract natural frequency (OMEGA, in Hz), damping ratio (DAMP),
C             and velocity (VEL, in m/s) for each mode from the complex
C             eigenvalues.
```

216

```
C
          DO 160 II = 1,NMODES
             OMEGA(II) = 0.
             DAMP(II) = 0.
             IF (ZR(II).NE.0.) THEN
                OMEGA(II) = 1./SQRT(ABS(ZR(II)))*ZR(II)/ABS(ZR(II))
                DAMP(II) = ZI(II)/ZR(II)
             ENDIF
             VEL(II) = (CHORD/2.)*OMEGA(II)/FREQ
             OMEGA(II) = OMEGA(II)/(2.*PI)
             IP(II) = II
160       CONTINUE
C
C         Sort multiple solutions by increasing frequency.
C
          DO 170 II = 1,(NMODES-1)
          DO 170 JJ = (II+1),NMODES
             IF (OMEGA(IP(JJ)) .LT. OMEGA(IP(II))) THEN
                IDUM = IP(JJ)
                IP(JJ) = IP(II)
                IP(II) = IDUM
             ENDIF
170       CONTINUE
C
          DO 180 II = 1,NMODES
             JJ = IP(II)
             WRITE(2,'(3(1PE11.3),$)') VEL(JJ),DAMP(JJ),OMEGA(JJ)
180       CONTINUE
          WRITE(2,*) ' '
C
999   CONTINUE
      CLOSE(2)
C
      STOP
      END

C-----FILE: AEROF.FOR --------------------------------------------------
C
      SUBROUTINE AEROF(LM,THETA,HBAR,VEL,FREQ,LPRINT,D,CZ)
C
C     Subroutine to calculate unsteady, non-linear, oscillatory aero-
C     dynamic coefficients by Fourier decomposition of the oscil-
C     latory, non-linear, stalled static aerodynamic force coefficient.
C
C     INPUT  VARIABLES: LM      = indicator for lift coefficient (LM='L')
C                                 or for moment coefficient (LM='M')
C                       THETA   = oscillating components of angle of
C                                 attack (rad)
C                       HBAR    = oscillating components of 1/4-chord
C                                 deflection (non-dimensional)
C                       VEL     = velocity (m/s)
C                       FREQ    = reduced frequency (non-dimensional)
C                       LPRINT  = logical print variable
C     OUTPUT VARIABLES: D       = coeffs of deviation from linear lift
C                                 curve in PHI domain (non-dimensional)
C                       CZ      = oscillating components of the desired
C                                 force coefficient (non-dimensional)
C
      INCLUDE     PARAM.INC
      INCLUDE     GLBBLK.INC
      CHARACTER   LM*1
      REAL        THETA(3),HBAR(3),VEL,FREQ,CZ(5)
      LOGICAL     LPRINT
C
```

217

```
C         *** Constants used in non-linear equations.
C
C         ALFA: Oscillating components of effective angle of attack (rad)
C         ALF0: Mean of effective angle of attack (rad)
C         ALFV: Amplitude of oscillation of effective angle of attack (rad)
C         TC:   Real angle of attack corrected for finite span (rad)
C         S,KV,LAM,SIG,ALF,W,D,E: Coefficients of ODE's (non-dim)
C
          REAL        ALFA(3),ALF0,ALFV,TC(3),S,KV,LAM,SIG,ALF
          CHARACTER   ANSWER*1
C         *** Variables used in linear calculations.
          REAL        LS,LC,CZ1(3)
          COMMON      / CZ1BLK / CZ1(3)
C         *** Variables used in non-linear calculations.
          INTEGER     LOREG,HIREG
          REAL        PHI(-MAXREG:MAXREG),BB(-MAXREG:MAXREG,0:MAXPOW)
          REAL        JCK,SINT(-MAXREG:MAXREG,0:MAXPOW),D(0:2)
          REAL        AA,RR,EE,B1,B2,B3,AMAT(5,5),BVEC(5)
          REAL        CP2(5),CZ2(5),K1,K2,K3,K4,DCZ(5)
C
          RE = RHOA*VEL*CHORD/RMUA
C
          IF (LPRINT) THEN
             OPEN (UNIT=3,FILE='AEROF.OUT',STATUS='NEW',FORM='FORMATTED')
             WRITE(3,*) ' '
             IF (LM.EQ.'L') WRITE(3,*) 'LIFT TRIAL USING AEROF SUBROUTINE'
             IF (LM.EQ.'M') WRITE(3,*) 'MOMENT TRIAL USING AEROF '//
         &      'SUBROUTINE'
             WRITE(3,*) ' '
             WRITE(3,*) 'INPUT VARIABLES:'
             WRITE(3,*) '================'
             WRITE(3,*) 'Reynold''s Number =',RE
             WRITE(3,*) 'THETA0 =',(THETA(1)*180./PI),' degs'
             WRITE(3,*) 'THETAs =',(THETA(2)*180./PI),' degs'
             WRITE(3,*) 'THETAc =',(THETA(3)*180./PI),' degs'
             WRITE(3,*) ' HBAR0 =',HBAR(1)
             WRITE(3,*) ' HBARs =',HBAR(2)
             WRITE(3,*) ' HBARc =',HBAR(3)
             WRITE(3,*) '  FREQ =',FREQ
          ENDIF
C
C         Calculate the perceived angle of attack coefficients [ALFA(i)],
C         the mean and vibratory amplitudes [ALF0 and ALFV], and the
C         phase [ZETA].
C
          ALFA(1) = THETA(1)
          ALFA(2) = THETA(2) + FREQ*HBAR(3)
          ALFA(3) = THETA(3) - FREQ*HBAR(2)
C
          ALF0 = ALFA(1)
          ALFV = SQRT(ALFA(2)**2+ALFA(3)**2)
          IF (ALFV.EQ.0.) ZETA=0.
          IF (ALFV.NE.0.) ZETA=ATAN2(ALFA(3),ALFA(2))
C
          IF (LPRINT) THEN
             WRITE(3,*) ' '
             WRITE(3,*) 'ALPHA0 =',(ALF0*180./PI),' degs'
             WRITE(3,*) 'ALPHAs =',(ALFA(2)*180./PI),' degs'
             WRITE(3,*) 'ALPHAc =',(ALFA(3)*180./PI),' degs'
             WRITE(3,*) 'ALPHAv =',(ALFV*180./PI),' degs'
             WRITE(3,*) ' ZETA  =',(ZETA*180./PI),' degs'
          ENDIF
C
C         Correct effective angle of attack and real angle of attack
```

218

```
C       for finite span.
C
        IF (REDUC) THEN
            ALF0 = ALF0/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
            ALFV = ALFV/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
            ALFA(1) = ALFA(1)/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
            ALFA(2) = ALFA(2)/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
            ALFA(3) = ALFA(3)/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
            TC(1) = THETA(1)/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
            TC(2) = THETA(2)/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
            TC(3) = THETA(3)/(1.+SLOPE('L')/PI/(2.*LENGTH/CHORD))
        ELSE
            TC(1) = THETA(1)
            TC(2) = THETA(2)
            TC(3) = THETA(3)
        ENDIF
C
        CZ1(1) = SLOPE(LM)*ALF0
C
C       Calculate lowest and highest region in which the alpha
C       oscillation passes through.
C
        LOREG = 0
        HIREG = 0
        AMIN = ALF0 - ALFV
        AMAX = ALF0 + ALFV
        DO 10 I = 1,IREGS(FOIL)
            IF ((TD(I).LE.AMIN).AND.(AMIN.LT.TD(I+1))) LOREG=I
            IF ((TD(I).LT.AMAX).AND.(AMAX.LE.TD(I+1))) HIREG=I
            IF ((-TD(I+1).LT.AMIN).AND.(AMIN.LE.-TD(I))) LOREG=-I
            IF ((-TD(I+1).LE.AMAX).AND.(AMAX.LT.-TD(I))) HIREG=-I
10      CONTINUE
C
        IF (.NOT.STEADY) THEN
C
C           Calculate coefficients of the linear differential equations.
C
            CALL COEFS_LIN(LM,ALF0,S,KV,LAM,SIG,ALF)
            IF (LPRINT) THEN
                WRITE(3,*) ' '
                WRITE(3,*) '  S =',S,'1/rad'
                WRITE(3,*) ' KV =',KV,'1/rad'
                WRITE(3,*) 'LAM =',LAM
                WRITE(3,*) 'SLP =',SLOPE(LM),'1/rad'
                WRITE(3,*) 'SIG =',SIG,'1/rad'
                WRITE(3,*) 'ALF =',ALF
            ENDIF
C
C           Calculate variables of linear aerodynamic equation.
C
            LS = SLOPE(LM)*ALFA(2)-SIG*FREQ*TC(3)
            LC = SLOPE(LM)*ALFA(3)+SIG*FREQ*TC(2)
            IF (LPRINT) THEN
                WRITE(3,*) ' '
                WRITE(3,*) 'Ls =',LS
                WRITE(3,*) 'Lc =',LC
            ENDIF
C
C           Calculate oscillatory contributions of linear aerodynamics.
C
            CZ1(2) = ((LAM*LAM+ALF*FREQ*FREQ)*LS+LAM*FREQ*
     &          (1.-ALF)*LC)/(LAM*LAM+FREQ*FREQ)
            CZ1(3) = ((LAM*LAM+ALF*FREQ*FREQ)*LC-LAM*FREQ*
     &          (1.-ALF)*LS)/(LAM*LAM+FREQ*FREQ)
```

```
          ENDIF
C
          IF (LPRINT) THEN
              WRITE(3,*) 'C'//LM//'1o =',CZ1(1)
              WRITE(3,*) 'C'//LM//'1s =',CZ1(2)
              WRITE(3,*) 'C'//LM//'1c =',CZ1(3)
          ENDIF
C
C         Calculate the coefficients of CZ2 in time: 1-constant,
C         2-first harmonic sine, 3-first harmonic cosine, 4-second
C         harmonic sine, 5-second harmonic cosine.
C
          IF (((LOREG.EQ.0).AND.(HIREG.EQ.0)).OR.(LINEAR)) THEN
C
C             Set coefficients equal to zero if oscillation
C             never enters the stalled regime or if only considering
C             the linear problem.
C
              DO 20 I = 1,5
                 CZ2(I) = 0.
20            CONTINUE
          ELSEIF ((STEADY).OR.(ALFV.EQ.0.)) THEN
C
C             If steady, calculate steady non-linear coefficient and set
C             unsteady non-linear coefficients to zero.
C
              CZ2(1) = -DCZS(LM,0,ALF0)
              DO 30 I = 2,5
                 CZ2(I) = 0.
30            CONTINUE
          ELSE
C
C             Calculate limits of integration for each region for
C             use in the Fourier analysis.
C
              PHI(LOREG) = -PI/2.
              PHI(HIREG+1) = PI/2.
              IF (LOREG.NE.HIREG) THEN
              DO 40 I = LOREG+1,HIREG
                 IF (I.LE.0) THEN
                    PHI(I) = ASIN((-TD(1-I)-ALF0)/ALFV)
                 ELSE
                    PHI(I) = ASIN((TD(I)-ALF0)/ALFV)
                 ENDIF
40            CONTINUE
              ENDIF
C
              IF (LPRINT) THEN
                 WRITE(3,*) ' '
                 DO 50 I = LOREG,HIREG+1
                    WRITE(3,*) 'REGION =',I,' PHI =',(PHI(I)*180./PI),
     &                 ' degs'
50            CONTINUE
              ENDIF
C
C             Calculate the coefficients of the polynomial expansion
C             sine series in each region that the oscillation passes thru.
C
              DO 130 I = LOREG,HIREG
                 IF (I.EQ.0) GOTO 130
C
C                Calculate constant coefficient.
C
                 BB(I,0) = REAL(SIGN(1,I))*DCZS(LM,0,TD(ABS(I)))
```

220

```fortran
            DO 60 J = 1,JMAX(ABS(I))
               BB(I,0) = BB(I,0) + REAL(SIGN(1,I)**(J+1))*
     &         A(LM,ABS(I),J)*(ALF0-REAL(SIGN(1,I))*TD(ABS(I)))**J
60          CONTINUE
C
C           Calculate higher order coefficients.
C
            DO 90 KK = 1,JMAX(ABS(I))
               BB(I,KK) = REAL(SIGN(1,I))**(KK+1)*A(LM,ABS(I),KK)*
     &            (ALFV**KK)
               IF (KK.NE.JMAX(ABS(I))) THEN
               DO 80 J = KK+1,JMAX(ABS(I))
C
C                 Calculate J-choose-KK.
C
                  JCK = 1.
                  DO 70 L = 1,KK
                     JCK = JCK*REAL(J-L+1)/REAL(L)
70                CONTINUE
C
C                 Add contribution of j-th power to bb(i,kk).
C
                  BB(I,KK) = BB(I,KK) + JCK*(SIGN(1,I))**(J+1)*
     &            A(LM,ABS(I),J)*(ALFV**KK)*((ALF0-REAL(SIGN(1,I))*
     &            TD(ABS(I)))**(J-KK))
80             CONTINUE
               ENDIF
90          CONTINUE
C
            IF (LPRINT) THEN
               WRITE(3,*) ' '
               WRITE(3,*) 'REGION =',I
               DO 100 KK = 0,JMAX(ABS(I))
                  WRITE(3,*) 'B(',KK,') =',BB(I,KK)
100            CONTINUE
            ENDIF
C
C           Calculate the integrals of the sine powers in each region
C           using Eqn 299 from "CRC Standard Math Tables," 28th edition.
C
            SINT(I,0) = PHI(I+1) - PHI(I)
            SINT(I,1) = COS(PHI(I)) - COS(PHI(I+1))
            DO 110 KK = 2,JMAX(ABS(I))+2
               SINT(I,KK) = (COS(PHI(I))*SIN(PHI(I))**(KK-1)-
     &         COS(PHI(I+1))*SIN(PHI(I+1))**(KK-1))/REAL(KK)+
     &         REAL(KK-1)/REAL(KK)*SINT(I,KK-2)
110         CONTINUE
C
            IF (LPRINT) THEN
               WRITE(3,*) ' '
               DO 120 KK = 0,JMAX(ABS(I))+2
                  WRITE(3,*) 'SINTEGRAL(',KK,') =',SINT(I,KK)
120            CONTINUE
            ENDIF
130      CONTINUE
C
C        Calculate the polynomial coefficients of the
C        Fourier expansion in the PHI domain.
C
         DCZ0 = 0.
         DCZ1 = 0.
         DCZ2 = 0.
         DO 150 I = LOREG,HIREG
            IF (I.NE.0) THEN
```

```
                   DO 140 KK = 0,JMAX(ABS(I))
                       DCZ0 = DCZ0 + BB(I,KK)*SINT(I,KK)/PI
                       DCZ1 = DCZ1 + BB(I,KK)*SINT(I,KK+1)*2./PI
                       DCZ2 = DCZ2 + BB(I,KK)*(SINT(I,KK)-2.*
         &                 SINT(I,KK+2))*2./PI
 140               CONTINUE
                   ENDIF
 150           CONTINUE
       C
               IF (LPRINT) THEN
                   WRITE(3,*) ' '
                   WRITE(3,*) 'DC'//LM//'0 =',DCZ0
                   WRITE(3,*) 'DC'//LM//'V1 =',DCZ1
                   WRITE(3,*) 'DC'//LM//'V2 =',DCZ2
               ENDIF
       C
               IF (LM.EQ.'L') THEN
                   D(0) = DCZ0
                   D(1) = DCZ1
                   D(2) = DCZ2
               ENDIF
       C
       C       Calculate coefficients of the non-linear aerodynamic
       C       differential equations. NOTE: this depends on DCL
       C       components - D(0), D(1), & D(2) - having already
       C       been calculated, i.e. that the calculations for
       C       LM='L' are done before LM='M'.
       C
               CALL COEFS_NON(ALF0,D(0),LM,AA,RR,EE,B1,B2,B3)
               IF (LPRINT) THEN
                   WRITE(3,*) ' '
                   IF ((B1.NE.0.).OR.(B2.NE.0.).OR.(B3.NE.0.)) THEN
                       WRITE(3,*) 'a0 =',AA*(1.-B1*D(0)**2)
                       WRITE(3,*) 'a1 =',AA*B1
                       WRITE(3,*) 'r0 =',SQRT(RR)*(1.-B2*D(0)**2)
                       WRITE(3,*) 'r1 =',SQRT(RR)*B2
                       WRITE(3,*) 'e0 =',EE*(1.-B3*D(0)**2)
                       WRITE(3,*) 'e1 =',EE*B3
                   ELSE
                       WRITE(3,*) 'A =',AA
                       WRITE(3,*) 'R =',RR
                       WRITE(3,*) 'E =',EE
                   ENDIF
               ENDIF
       C
               AMAT(1,1) = RR * (1. + B2 * (D(1)**2+D(2)**2) +
         &       B2**2 * (2.*D(0)**2*D(1)**2 + .375*D(1)**4 -
         &       3.*D(0)*D(1)**2*D(2) + 2.*D(0)**2*D(2)**2 +
         &       1.5*D(1)**2*D(2)**2 + .375*D(2)**4))
               AMAT(1,2) = RR * (B2 * (2.*D(0)*D(1)-D(1)*D(2)) +
         &       B2**2 * (1.5*D(0)*D(1)**3 - 2.*D(0)**2*D(1)*
         &       D(2) - D(1)**3*D(2) + 3.*D(0)*D(1)*D(2)**2 -
         &       .75*D(1)*D(2)**3))
               AMAT(1,3) = -FREQ*AA*B1 * (D(0)*D(1) -
         &       .5*D(1)*D(2))
               AMAT(1,4) = 2.*FREQ*AA*B1 * (-.25*D(1)**2 +
         &       D(0)*D(2))
               AMAT(1,5) = RR * (B2 * (-.5*D(1)**2+2.*D(0)*
         &       D(2)) + B2**2 * (-D(0)**2*D(1)**2 - .25*D(1)**4 +
         &       3.*D(0)*D(1)**2*D(2) - 1.125*D(1)**2*D(2)**2 +
         &       1.5*D(0)*D(2)**3))
       C
               AMAT(2,1) = RR * (B2 * (4.*D(0)*D(1) - 2.*D(1)*
         &       D(2)) + B2**2 * (3.*D(0)*D(1)**3 - 4.*D(0)**2*
```

222

```fortran
     &      D(1)*D(2) - 2.*D(1)**3*D(2) + 6.*D(0)*D(1)*
     &      D(2)**2 - 1.5*D(1)*D(2)**3))
          AMAT(2,2) = RR * (1. + B2 * (1.5*D(1)**2 -
     &      2.*D(0)*D(2) + D(2)**2) + B2**2 * (3.*D(0)**2*
     &      D(1)**2 + .625*D(1)**4 - 6.*D(0)*D(1)**2*D(2) +
     &      2.*D(0)**2*D(2)**2 + 2.625*D(1)**2*D(2)**2 -
     &      1.5*D(0)*D(2)**3 + .375*D(2)**4))
          AMAT(2,3) = -FREQ*AA * (1. + B1 * (.75*D(1)**2 -
     &      D(0)*D(2) + .5*D(2)**2))
          AMAT(2,4) = 2.*FREQ*AA*B1 * (-D(0)*D(1) +
     &      D(1)*D(2))
          AMAT(2,5) = RR * (B2 * (-2.*D(0)*D(1) + 2.*D(1)*
     &      D(2)) + B2**2 * (-2.*D(0)*D(1)**3 + 4.*D(0)**2*
     &      D(1)*D(2) + 1.75*D(1)**3*D(2) - 4.5*D(0)*D(1)*
     &      D(2)**2 + 1.5*D(1)*D(2)**3))
C

          AMAT(3,1) = 0.
          AMAT(3,2) = FREQ*AA * (1. + B1 * (.25*D(1)**2 +
     &      D(0)*D(2) + .5*D(2)**2))
          AMAT(3,3) = RR * (1. + B2 * (.5*D(1)**2 +
     &      2.*D(0)*D(2) + D(2)**2) + B2**2 * (D(0)**2*
     &      D(1)**2 + .125*D(1)**4 - 2.*D(0)**3*D(2) +
     &      2.*D(0)**2*D(2)**2 + 2.*D(0)**3*D(2) +
     &      .375*D(1)**2*D(2)**2 + 1.5*D(0)*D(2)**3 +
     &      .375*D(2)**4))
          AMAT(3,4) = RR * (B2 * 2.*D(0)*D(1) + B2**2 *
     &      (D(0)*D(1)**3 - .25*D(1)**3*D(2) + 1.5*D(0)*
     &      D(1)*D(2)**2))
          AMAT(3,5) = -2.*FREQ*AA*B1 * D(0)*D(1)
C

          AMAT(4,1) = 0.
          AMAT(4,2) = FREQ*AA*B1 * D(0)*D(1)
          AMAT(4,3) = RR * (B2 * 2.*D(0)*D(1) + B2**2 *
     &      (D(0)*D(1)**3 - .25*D(1)**3*D(2) + 1.5*D(0)*
     &      D(1)*D(2)**2))
          AMAT(4,4) = RR * (1. + B2 * (D(1)**2 +
     &      .5*D(2)**2) + B2**2 * (2.*D(0)**2*D(1)**2 +
     &      .3125*D(1)**4 - 1.5*D(0)*D(1)**2*D(2) +
     &      D(0)**2*D(2)**2 + .75*D(1)**2*D(2)**2 +
     &      .125*D(2)**4))
          AMAT(4,5) = -2.*FREQ*AA * (1. + B1 *
     &      (.5*D(1)**2 + .25*D(2)**2))
C

          AMAT(5,1) = RR * (B2 * (D(1)**2 + 4.*D(0)*
     &      D(2)) + B2**2 * (-4.*D(0)**2*D(1)**2 -
     &      .5*D(1)**4 + 6.*D(0)*D(1)**2*D(2) -
     &      2.25*D(1)**2*D(2)**2 + 3.*D(0)*D(2)**3))
          AMAT(5,2) = RR * (B2 * (-2.*D(0)*D(1) + 2.*
     &      D(1)*D(2)) + B2**2 * (-2.*D(0)*D(1)**3 +
     &      4.*D(0)**2*D(1)*D(2) + 1.75*D(1)**3*D(2) -
     &      4.5*D(0)*D(1)*D(2)**2 + 1.5*D(1)*D(2)**3))
          AMAT(5,3) = -FREQ*AA*B1 * (D(0)*D(1) + D(1)*D(2))
          AMAT(5,4) = 2.*FREQ*AA * (1. + B1 * (.5*D(1)**2 +
     &      .75*D(2)**2))
          AMAT(5,5) = RR * (1. + B2 * (D(1)**2 + 1.5*
     &      D(2)**2) + B2**2 * (2.*D(0)**2*D(1)**2 +
     &      .4375*D(1)**4 - 4.5*D(0)*D(1)**2*D(2) +
     &      3.*D(0)**2*D(2)**2 + 2.25*D(1)**2*D(2)**2 +
     &      .625*D(2)**4))
C

          BVEC(1) = -(AMAT(1,1)*DCZ0 + AMAT(1,2)*DCZ1 +
     &      AMAT(1,5)*DCZ2)
          BVEC(2) = -(AMAT(2,1)*DCZ0 + AMAT(2,2)*DCZ1 +
     &      AMAT(2,5)*DCZ2)
```

```
        BVEC(3) = -(FREQ*DCZ1 * EE * (1. + B3 *
     &     (.25*D(1)**2 + D(0)*D(2) + .5*D(2)**2)) -
     &     2.*FREQ*DCZ2 * EE*B3 * D(0)*D(1))
        BVEC(4) = -(FREQ*DCZ1 * EE*B3 * D(0)*D(1) -
     &     2.*FREQ*DCZ2 * EE * (1. + B3 * (.5*D(1)**2 +
     &     .25*D(2)**2)))
        BVEC(5) = -(AMAT(5,1)*DCZ0 + AMAT(5,2)*DCZ1 +
     &     AMAT(5,5)*DCZ2)
C
        AMAT(2,2) = AMAT(2,2) - FREQ**2
        AMAT(3,3) = AMAT(3,3) - FREQ**2
        AMAT(4,4) = AMAT(4,4) - 4.*FREQ**2
        AMAT(5,5) = AMAT(5,5) - 4.*FREQ**2
C
160     FORMAT(' |',5(1PE10.2),'|',A7,'|',1PE10.2,'|')
        IF (LPRINT) THEN
           WRITE(3,*) ' '
           DO 170 JJ = 1,5
              IF (JJ.NE.3) THEN
                 WRITE(3,160) (AMAT(JJ,KK),KK=1,5),
     &              '        ',BVEC(JJ)
              ELSE
                 WRITE(3,160) (AMAT(JJ,KK),KK=1,5),
     &              '.C'//LM//'2 = ',BVEC(JJ)
              ENDIF
170        CONTINUE
        ENDIF
C
        CALL SOLVE(AMAT,BVEC,CP2,5,1,5)
        IF (LPRINT) THEN
           WRITE(3,*) ' '
           WRITE(3,*) 'Nonlinear coefficients in Phi '//
     &        'or (Omega*Tau+Zeta) domain:'
           WRITE(3,'(/5(A,1PE10.3))') ' C'//LM//'2P0=',CP2(1),
     &        ' ; C'//LM//'2Ps1=',CP2(2),' ; C'//LM//'2Pc1=',CP2(3),
     &        ' ; C'//LM//'2Ps2=',CP2(4),' ; C'//LM//'2Pc2=',CP2(5)
        ENDIF
C
C       Convert nonlinear coefficients to normal OMEGA*TAU domain.
C
        CZ2(1) = CP2(1)
        CZ2(2) = CP2(2)*COS(ZETA) - CP2(3)*SIN(ZETA)
        CZ2(3) = CP2(3)*COS(ZETA) + CP2(2)*SIN(ZETA)
        CZ2(4) = CP2(4)*COS(2.*ZETA) - CP2(5)*SIN(2.*ZETA)
        CZ2(5) = CP2(5)*COS(2.*ZETA) + CP2(4)*SIN(2.*ZETA)
C
C       If diagnostics in effect, compare against old analysis,
C       from Dunn Master's thesis, using constant coefficients.
C
        IF (LPRINT) THEN
C
           DCZ(1) = D(0)
           DCZ(2) = D(1)*COS(ZETA)
           DCZ(3) = D(1)*SIN(ZETA)
           DCZ(4) = -D(2)*SIN(2.*ZETA)
           DCZ(5) = D(2)*COS(2.*ZETA)
C
           CP2(1) = -DCZ(1)
C
C          Calculate first harmonic coefficients of unsteady
C          aerodynamics.
C
           K1 = RR-FREQ**2
           K2 = AA*FREQ
```

```fortran
          K3 = -RR*DCZ(2)+EE*FREQ*DCZ(3)
          K4 = -RR*DCZ(3)-EE*FREQ*DCZ(2)
          CP2(2) = (K1*K3+K2*K4)/(K1*K1+K2*K2)
          CP2(3) = (K1*K4-K2*K3)/(K1*K1+K2*K2)
C
          WRITE(3,'(/A)') ' First Harmonic in Omega*Tau domain'
          WRITE(3,*) 'DC'//LM//'0  =',DCZ(1)
          WRITE(3,*) 'DC'//LM//'s1 =',DCZ(2)
          WRITE(3,*) 'DC'//LM//'c1 =',DCZ(3)
          WRITE(3,*) 'K1(1) =',K1
          WRITE(3,*) 'K2(1) =',K2
          WRITE(3,*) 'K3(1) =',K3
          WRITE(3,*) 'K4(1) =',K4
          WRITE(3,*) 'CZ2o  =',CP2(1)
          WRITE(3,*) 'CZ2s1 =',CP2(2)
          WRITE(3,*) 'CZ2c1 =',CP2(3)
C
C         Calculate second harmonic coefficients of unsteady
C         aerodynamics.
C
          K1 = RR-(2.*FREQ)**2
          K2 = AA*(2.*FREQ)
          K3 = -RR*DCZ(4)+EE*2.*FREQ*DCZ(5)
          K4 = -RR*DCZ(5)-EE*2.*FREQ*DCZ(4)
          CP2(4) = (K1*K3+K2*K4)/(K1*K1+K2*K2)
          CP2(5) = (K1*K4-K2*K3)/(K1*K1+K2*K2)
C
          WRITE(3,'(/A)') ' Second Harmonic in Omega*Tau domain'
          WRITE(3,*) 'DC'//LM//'s2 =',DCZ(4)
          WRITE(3,*) 'DC'//LM//'c2 =',DCZ(5)
          WRITE(3,*) 'K1(2) =',K1
          WRITE(3,*) 'K2(2) =',K2
          WRITE(3,*) 'K3(2) =',K3
          WRITE(3,*) 'K4(2) =',K4
          WRITE(3,*) 'CZ2s2 =',CP2(4)
          WRITE(3,*) 'CZ2c2 =',CP2(5)
        ENDIF
      ENDIF
C
      IF (LPRINT) THEN
        WRITE(3,*) ' '
        WRITE(3,*) 'C'//LM//'20  =',CZ2(1)
        WRITE(3,*) 'C'//LM//'2s1 =',CZ2(2)
        WRITE(3,*) 'C'//LM//'2c1 =',CZ2(3)
        WRITE(3,*) 'C'//LM//'2s2 =',CZ2(4)
        WRITE(3,*) 'C'//LM//'2c2 =',CZ2(5)
      ENDIF
C
C     Add apparent mass terms.
C
      CZ1(2) = CZ1(2) - S*FREQ*ALFA(3) - KV*FREQ*FREQ*TC(2)
      CZ1(3) = CZ1(3) + S*FREQ*ALFA(2) - KV*FREQ*FREQ*TC(3)
C
C     Combine linear and non-linear terms for
C     total coefficients of full non-linear aerodynamics.
C
      CZ(1) = CZ1(1) + CZ2(1)
      CZ(2) = CZ1(2) + CZ2(2)
      CZ(3) = CZ1(3) + CZ2(3)
      CZ(4) = CZ2(4)
      CZ(5) = CZ2(5)
C
      IF (LPRINT) THEN
        WRITE(3,*) ' '
```

```
          WRITE(3,*) 'C'//LM//'0  =',CZ(1)
          WRITE(3,*) 'C'//LM//'s1 =',CZ(2)
          WRITE(3,*) 'C'//LM//'c1 =',CZ(3)
          WRITE(3,*) 'C'//LM//'s2 =',CZ(4)
          WRITE(3,*) 'C'//LM//'c2 =',CZ(5)
          CLOSE(3)
          WRITE(*,*) 'CZ0 =',CZ(1),' ; CZs1 =',CZ(2),' ; CZc1 =',CZ(3),
     &      ' ; CZs2 =',CZ(4),' ; CZc2 =',CZ(5)
          PAUSE
        ENDIF
C
        RETURN
        END

C-----FILE: CHARAC.FOR -------------------------------------------------
C
C       Subroutines and functions which describe the static lift curve
C       of the desired airfoil ('OA212' for the OA212 or 'NAC12' for the
C       NACA-0012).
C
C======================================================================
C
C       Function to describe the slope of the linear part of the lift
C       curve.
C
        REAL FUNCTION SLOPE(LM)
C
        INCLUDE     PARAM.INC
        INCLUDE     GLBBLK.INC
        CHARACTER   LM*1
C
C       OA212 lift slope taken from Rogers, "Applications of an
C       Analytic Stall Model to Time-History and Eigenvalue Analysis
C       of Rotor Blades", Journal of the American Helicopter Society,
C       January 1984.
C
        IF ((FOIL.EQ.'OA212').AND.(LM.EQ.'L')) SLOPE=7.1
C
C       ** NACA-0012 LIFT SLOPE APPROXIMATED FROM NACA REPORT 586,
C       ** JACOBS & SHERMAN, FIGURE 3
C
        IF ((FOIL.EQ.'NAC12').AND.(LM.EQ.'L')) SLOPE=0.103*(180./PI)
C       IF ((FOIL.EQ.'NAC12').AND.(LM.EQ.'L')) SLOPE=2.*PI
C
C       ** NACA-0012 MOMENT SLOPE APPROXIMATED FROM NACA TM-84245-VOL-2,
C       ** McALISTER, PUCCI, McCROSKEY, AND CARR, FIGURE 9
C
        IF ((FOIL.EQ.'NAC12').AND.(LM.EQ.'M')) SLOPE=0.002*(180./PI)
        IF ((FOIL.EQ.'NAC12').AND.(LM.EQ.'M')) SLOPE=0.
        RETURN
        END
C
C======================================================================
C
C       Function to describe the DERIV-th derivative of Delta-CZ Static
C       (DCZS, the static deviation from the static, linear lift curve),
C       evaluated at angle THETA.
C
        REAL FUNCTION DCZS(LM,IDERIV,THETA)
C
        INCLUDE     PARAM.INC
        INCLUDE     GLBBLK.INC
        CHARACTER   LM*1
C
```

226

```fortran
C        Find region in which THETA lies.
C
         IREG = 0
         DO 10 I = 1,IREGS(FOIL)
            IF ((TD(I).LT.THETA).AND.(THETA.LE.TD(I+1))) IREG=I
            IF ((-TD(I+1).LE.THETA).AND.(THETA.LT.-TD(I))) IREG=-I
10       CONTINUE
C
         IF (LINEAR) THEN
            DCZS = 0.
         ELSEIF (IDERIV.EQ.0) THEN
C
C        Calculate zero-th derivative.
C
            IF (IREG.EQ.0) THEN
               DCZS = 0.
            ELSE
               DO 30 I = 0,ABS(IREG)-1
C
C                 Calculate DCZS at TD(IREG) by calculating region
C                 by region DCZS(TD(I)) in each region previous to IREG.
C
                  IF (I.EQ.0) THEN
                     DCZS = 0.
                  ELSE
                     DO 20 J = 1,JMAX(I)
                        DCZS = DCZS + A(LM,I,J)*(TD(I+1)-TD(I))**J
20                   CONTINUE
                  ENDIF
30             CONTINUE
C
C              Calculate DCZS(THETA) using the previously calculated
C              DCZS(TD(IREG)) as a starting point and the power
C              expansion of DCZS in region IREG.
C
               DO 40 J = 1,JMAX(ABS(IREG))
                  DCZS = DCZS + A(LM,ABS(IREG),J)*(ABS(THETA)-
     &               TD(ABS(IREG)))**J
40             CONTINUE
               DCZS = DCZS*REAL(SIGN(1,IREG))
            ENDIF
C
         ELSEIF (IDERIV.GT.0) THEN
C
C        Calculate higher derivatives.
C
            DCZS = 0.
            IF ((IREG.NE.0).AND.(IDERIV.LE.JMAX(ABS(IREG)))) THEN
               DO 60 J = IDERIV,JMAX(ABS(IREG))
C
C                 Calculate J!/(J-IDERIV)!.
C
                  IFAC = 1
                  DO 50 JJ = J-IDERIV+1,J
                     IFAC = IFAC*JJ
50                CONTINUE
C
C                 Add contribution of J-th power, differentiated IDERIV
C                 times, to the overall derivative.
C
                  DCZS = DCZS + A(LM,ABS(IREG),J)*REAL(IFAC)*
     &               (ABS(THETA)-TD(ABS(IREG)))**(J-IDERIV)
60             CONTINUE
            ENDIF
```

```
          DCZS = DCZS*REAL(SIGN(1,IREG))
       ENDIF
C
       RETURN
       END
C
C====================================================================
C
C       Function to describe number of regions into which the lift
C       curve is divided.
C
       INTEGER FUNCTION IREGS(FOIL)
C
       CHARACTER FOIL*5
C      IF (FOIL.EQ.'OA212') IREGS=2
       IF (FOIL.EQ.'OA212') IREGS=1
C      IF (FOIL.EQ.'NAC12') IREGS=3
       IF (FOIL.EQ.'NAC12') IREGS=2
       RETURN
       END
C
C====================================================================
C
C       Function to describe the angles at which each of the regions of
C       the lift curve begins [units of radians].
C
       REAL FUNCTION TD(IREG)
C
       INCLUDE       PARAM.INC
       INCLUDE       GLBBLK.INC
C
       IF (FOIL.EQ.'OA212') THEN
          IF (IREG.EQ.1) TD=10.*PI/180.
C         IF (IREG.EQ.2) TD=24.8*PI/180.
C         IF (IREG.GE.3) TD=PI/2.
          IF (IREG.EQ.2) TD=PI/2.
       ELSEIF (FOIL.EQ.'NAC12') THEN
C
C         ** PARABOLIC, STALL, AND STRAIGHT LINE ANGLES FOR NACA-0012
C         ** APPROXIMATED USING FIT TO LOG(RE) DATA FROM NACA REPORT 586,
C         ** JACOBS & SHERMAN, FIGURE 3.
C
          RVAL = LOG(RE/3.4E5)/LOG(2.)
C
C         Parabolic from half of stall to stall angle (11 deg),
C         exponential decay to large angles (25 deg), flat line above.
C
C         IF (IREG.EQ.1) TD=(11.+2.143*RVAL)/2.*PI/180.
C         IF ((IREG.EQ.1).AND.(RE.LT.3.4D5)) TD=11./2.*PI/180.
C         IF (IREG.EQ.2) TD=(11.+2.143*RVAL)*PI/180.
C         IF ((IREG.EQ.2).AND.(RE.LT.3.4D5)) TD=11.*PI/180.
C         IF (IREG.EQ.3) TD=25.*PI/180.
C         IF (IREG.GE.4) TD=PI/2.
C
C         Straight line to stall angle (11 deg), exponential decay to
C         large angles (25 deg), flat line above.
C
C         IF (IREG.EQ.1) TD=(11.+2.143*RVAL)*PI/180.
C         IF ((IREG.EQ.1).AND.(RE.LT.3.4D5)) TD=11.*PI/180.
C         IF (IREG.EQ.2) TD=25.*PI/180.
C         IF (IREG.GE.3) TD=PI/2.
C
C         Slight drop after stall angle (8 deg), then flat line at
C         high angles (>20 deg)
```

228

```fortran
C
          IF (IREG.EQ.1) TD=8.*PI/180.
          IF (IREG.EQ.2) TD=20.*PI/180.
          IF (IREG.EQ.3) TD=PI/2.
C
C         Flat line lift curve after stall angle.
C
C         CLASY = .75 + .0536*RVAL
C         IF (RVAL.LE.0) CLASY=.75
C         IF (IREG.EQ.1) TD=CLASY/SLOPE('L')
C         IF (IREG.EQ.2) TD=PI/2.
C
      ENDIF
      RETURN
      END
C
C==================================================================
C
C     Function to describe the maximum power of the polynomial
C     approximation used in region IREG.
C
      INTEGER FUNCTION JMAX(IREG)
C
      INCLUDE       PARAM.INC
      INCLUDE       GLBBLK.INC
C
      IF (FOIL.EQ.'OA212') THEN
C
C         Straight lines connecting each region.
C
          JMAX=1
C
C         IF (IREG.EQ.1) JMAX=7
C         IF (IREG.EQ.2) JMAX=1
C
      ELSEIF (FOIL.EQ.'NAC12') THEN
C
C         Straight lines connecting each region.
C
          JMAX=1
C
C         Parabolic below stall, exponential to asymptotic, level off
C         to flat line for high angles.
C
C         IF (IREG.EQ.1) JMAX=2
C         IF (IREG.EQ.2) JMAX=10
C         IF (IREG.EQ.3) JMAX=1
C
      ENDIF
      RETURN
      END
C
C==================================================================
C
C     Function to prescribe the coefficients of the polynomial
C     approximation to Delta-CZ in region IREG. Powers of (180/PI)
C     are present because of conversions from units of degrees to
C     radians.
C
      REAL FUNCTION A(LM,IREG,J)
C
      INCLUDE       PARAM.INC
      INCLUDE       GLBBLK.INC
      CHARACTER     LM*1
```

```
C
                    IF ((FOIL.EQ.'OA212').AND.(LM.EQ.'L')) THEN
C                       IF (IREG.EQ.1) THEN
C                          IF (J.EQ.1) A=0.
C                          IF (J.EQ.2) A=+6.3059700D-2*(180./PI)**2
C                          IF (J.EQ.3) A=-1.3952010D-2*(180./PI)**3
C                          IF (J.EQ.4) A=+1.7390851D-3*(180./PI)**4
C                          IF (J.EQ.5) A=-1.2451913D-4*(180./PI)**5
C                          IF (J.EQ.6) A=+4.6849257D-6*(180./PI)**6
C                          IF (J.EQ.7) A=-7.0879730D-8*(180./PI)**7
C                       ELSEIF (IREG.EQ.2) THEN
C                          IF (J.EQ.1) A=SLOPE(LM)
C                       ENDIF
                        A = 0.
                        IF ((IREG.EQ.1).AND.(J.EQ.1)) A=SLOPE(LM)
                    ELSEIF (FOIL.EQ.'NAC12') THEN
C
C                       ** PARABOLIC, STALL, AND STRAIGHT LINE COEFFICIENTS FOR
C                       ** NACA-0012 LIFT SLOPE ARE APPROXIMATED USING FIT TO LOG(RE)
C                       ** DATA FROM NACA REPORT 586, JACOBS & SHERMAN, FIGURE 3.
C                       ** COEFFICIENTS FOR MOMENT SLOPE TAKEN FROM McALISTER, NASA
C                       ** TM-84245, FIGURE ?.
C
C                       RVAL = LOG(RE/3.4E5)/LOG(2.)
C
C                       Calculate the maximum lift/moment coefficient, dependent on
C                       the log of the Reynold's Number.
C
C                       CZMAX = .86 + .24*RVAL
C                       IF (RVAL.LE.0) CZMAX=.86+.03*RVAL
C                       IF (LM.EQ.'M') CZMAX=.04
C
C                       Calculate the asymptotic lift/moment coefficient (i.e. the
C                       lift/moment coefficient when the angle of attack tends to large
C                       angles), dependent on the log of the Reynold's Number.
C
C                       RVAL = 0.
C                       CZASY = .75 + .0536*RVAL
C                       IF (RVAL.LE.0) CZASY=.75
C                       IF (LM.EQ.'M') CZASY=-.12
C
C                       Calculate the coefficient o: the exponential decay from maximum
C                       to asymptotic, dependent or the log of the Reynold's Number.
C
C                       RNU = -.2-.07*RVAL
C                       IF (RVAL.LE.0) RNU=-.2
C
C                       Calculate the necessary polynomial coefficients for each
C                       region, using the previously calculated maximum, asymptotic,
C                       and exponential decay coefficients.
C
C                       A = 0.
C                       IF (LM.EQ.'L') THEN
C                          IF (IREG.EQ.1) THEN
C
C                             Parabolic fit from end of linear region (region 0) to
C                             point of maximum lift/moment (end of region 1). The
C                             polynomial coefficients are chosen such that the slope is
C                             continuous at the juncture of regions 0 and 1.
C
C                             IF (J.EQ.1) A=0.
C                             IF (J.EQ.2) A=(SLOPE(LM)*TD(2)-CZMAX)/(TD(2)-TD(1))**2
C
C                          ELSEIF (IREG.EQ.2) THEN
```

230

```fortran
C
C                      Exponential fit with decay coefficient RNU from end of
C                      parabolic region (region 1) to beginning of asymptotic
C                      region (region 3). The polynomial coefficients are chosen
C                      such that they fit a power series expansion of the
C                      exponential decay.
C
C                      IFAC = 1
C                      DO 10 I = 1,J
C                         IFAC = IFAC*I
C10                    CONTINUE
C                      A = (CZASY-CZMAX)*RNU**J/REAL(IFAC)*(180./PI)**J
C                      IF (J.EQ.1) A=A+SLOPE(LM)
C
C                  ELSEIF (IREG.EQ.3) THEN
C
C                      Flat line fit for the asymptotic region (region 3).
C
C                      IF (J.EQ.1) A=SLOPE(LM)
C                  ENDIF
C              ELSEIF (LM.EQ.'L') THEN
C
C                  Moment coefficient remains constant up to stall (i.e.
C                  through regions 0 & 1), straight line drop to the asymptotic
C                  value in region 2, and flat line afterward.
C
C                  IF ((IREG.EQ.2).AND.(J.EQ.1)) A=-CZASY/(TD(3)-TD(2))
C                  IF ((IREG.EQ.3).AND.(J.EQ.1)) A=SLOPE(LM)
C              ENDIF
C
          IF (LM.EQ.'L') THEN
              IF (IREG.EQ.1) A = (SLOPE(LM)*TD(2)-CZASY)/(TD(2)-TD(1))
              IF (IREG.EQ.2) A = SLOPE(LM)
          ELSEIF (LM.EQ.'M') THEN
              IF (IREG.EQ.1) A = (SLOPE(LM)*TD(2)-CZASY)/(TD(2)-TD(1))
              IF (IREG.EQ.2) A = SLOPE(LM)+(CZASY+.15)/
     &          (30.*PI/180.-TD(2))
          ENDIF
C
      ENDIF
      RETURN
      END

C-----FILE: COEFS.FOR -------------------------------------------------
C
C     Subroutine to calculate unsteady lift/moment coefficients
C     for linear (CZ1) equations.
C
      SUBROUTINE COEFS_LIN(LM,THETA,S,KV,LAM,SIG,ALPHA)
C
      INCLUDE      PARAM.INC
      INCLUDE      GLBBLK.INC
      CHARACTER    LM*1
      REAL         THETA,S,KV,LAM,SIG,ALPHA
C
      IF ((FOIL.EQ.'OA212').AND.(LM.EQ.'L')) THEN
C
C         Coefficients from Rogers, "Applications of an Analytic Stall
C         Model to Time-History and Eigenvalue Analysis of Rotor Blades",
C         Journal of the American Helicopter Society, January 1984,
C         page 26, equations (4) to (8).
C
          S = 5.
          KV = 0.
```

231

```fortran
              LAM = 0.2
              SIG = (SLOPE(LM) - 4.*(1.+1.43*DCZS(LM,0,THETA)))/LAM
              ALPHA = 0.
          ELSEIF (FOIL.EQ.'NAC12') THEN
              IF (LM.EQ.'L') THEN
C
C                 Coefficients from Petot, "Dynamic Stall Modeling of the
C                 NACA 0012 Profile", Short Note, page 58, equations (2)
C                 and (3), converted to same notation as Rogers (above).
C
C                 S = 0.09*(180./PI)
                  S = PI
                  KV = PI/2.
C                 LAM = 0.2
                  LAM = 0.15
C                 SIG = (0.08-0.13*DCL0)*(180./PI)/LAM
                  SIG = SLOPE(LM)
C                 ALPHA = .5
                  ALPHA = .55
              ELSEIF ((FOIL.EQ.'NAC12').AND.(LM.EQ.'M')) THEN
C                 S = -0.0304*180./PI
                  S = -PI/4.
                  KV = -3.*PI/16.
                  LAM = 0.
C                 SIG = 0.0089*180./PI
C                 IF ((THETA.GT.(13.*PI/180.)).AND.(.NOT.LINEAR))
C     &               SIG=SIG-.00067*180./PI*(THETA*180./PI-13.)/LAM
                  SIG = -PI/4.
                  ALPHA = 1.
C                 GAM = 0.16
C                 IF (THETA.GT.(13.*PI/180.)) GAM=GAM+.035*
C     &               (THETA*180./PI-13.)
C                 ALF = 1./11.5/GAM
C                 C = 3.5
C                 IF (THETA.GT.(15.6*PI/180.)) C=C+3./PI*ATAN(
C     &                  SQRT(3.)/1.2*(THETA*180./PI-15.6))
              ENDIF
          ENDIF
C
          RETURN
          END
C
C===============================================================================
C
C     Subroutine to calculate unsteady lift/moment coefficients
C     for non-linear (CZ2) equations.
C
      SUBROUTINE COEFS_NON(THETA,DCL0,LM,AA,RR,EE,B1,B2,B3)
C
      INCLUDE     PARAM.INC
      INCLUDE     GLBBLK.INC
      REAL        THETA,DCL0,AA,RR,EE,B1,B2,B3
      REAL        GAM,ALF,C,L1,L2,L3,L4,L5,L6
      CHARACTER*1 LM
      REAL        A0L,A1L,R0L,R1L,E0L,E1L
      REAL        A0M,A1M,R0M,R1M,E0M,E1M
      COMMON      / COEFBLK / A0L,A1L,R0L,R1L,E0L,E1L,
     &            A0M,A1M,R0M,R1M,E0M,E1M
C
      IF (ATYPE.EQ.0) THEN
C
C         Coefficients from Rogers, "Applications of an Analytic Stall
C         Model to Time-History and Eigenvalue Analysis of Rotor Blades",
C         Journal of the American Helicopter Society, January 1984,
```

```fortran
C            page 26, equations (4) to (8) for airfoil type OA212.
C
             GAM = 0.1
             IF (THETA.GT.(13.*PI/180.)) GAM=GAM+0.023*(THETA*
     &         (180./PI)-13.)
             ALF = 0.105/GAM
             C = 2.
             IF (THETA.GT.(13.*PI/180.)) C=C-5.1*ATAN(1.21*
     &         (THETA*(180./PI)-13.))
C
             AA = 2.*ALF*GAM
             RR = GAM**2*(1.+ALF**2)
             EE = RR*C
             B1 = 0.
             B2 = 0.
             B3 = 0.
          ELSE
             IF (ATYPE.EQ.1) THEN
C
C                Coefficients from Petot, "Dynamic Stall Modeling of the
C                NACA 0012 Profile," Short Note, Recherches Aerospatiales,
C                1984-6, pp. 55-58, with corrections for low Reynold's number
C                from Petot & Loiseau, "Succesive Smoothing Algorithm for
C                Constructing the Semi-Empirical Model Developed at ONERA to
C                Predict Unsteady Aerodynamic Forces," NASA TM-76681, March
C                1982, for airfoil type NACA 0012.
C
                L1 = .25
                L2 = .10
                IF (RE.LE.3.4E5) L2 = .40
                L3 = .20
                L4 = .10
                IF (RE.LT.3.4E5) L4 = .23
                L5 = 0.
                L6 = -.60
                IF (RE.LT.3.4E5) L6 = -2.7
             ELSEIF (LM.EQ.'L') THEN
                L1 = A0L
                L2 = A1L
                L3 = R0L
                L4 = R1L
                L5 = E0L
                L6 = E1L
             ELSEIF (LM.EQ.'M') THEN
                L1 = A0M
                L2 = A1M
                L3 = R0M
                L4 = R1M
                L5 = E0M
                L6 = E1M
             ENDIF
C
             AA = L1 + L2*DCL0**2
             RR = (L3 + L4*DCL0**2)**2
             EE = L5+L6*DCL0**2
             IF (ATYPE.LE.1) EE = RR*EE
C
             IF (LCONST) THEN
                B1 = 0.
                B2 = 0.
                B3 = 0.
             ELSE
                B1 = L2/AA
                B2 = L4/SQRT(RR)
```

```fortran
            B3 = L6/EE
            IF (ATYPE.LE.1) B3 = L3*(L3*
     &         L6+2.*L4*L5)/EE
          ENDIF
C
      ENDIF
C
      RETURN
      END

C-----FILE: CORREC.FOR --------------------------------------------
C
C     Functions to describe the spanwise and chordwise distributions
C     applicable to the 2-dimensional lift, moment, and drag
C     coefficients.
C
      REAL FUNCTION SC(XBAR)
      SC = 1.
      SC = 1.11*(1.-XBAR**9)
      RETURN
      END
C
      REAL FUNCTION CC(YBAR)
      CC = 3.*(0.5-YBAR)**2
      RETURN
      END

C-----FILE: DRDQ.FOR ----------------------------------------------
C
      SUBROUTINE R_DERIV(BEN_TOR,VEL,AOA,FREQ,QLIT,RES,RGEOM,DRDQ)
C
C     Subroutine to calculate the Jacobian matrix d(RES)/d(QLIT) by
C     numerical estimation of the derivatives.
C
C     INPUT VARIABLES:   BEN_TOR = bending/torsion flag
C                            VEL = velocity (m/s)
C                            AOA = root angle of attack (rad)
C                           FREQ = reduced frequency (non-dim)
C                           QLIT = modal amplitudes (m)
C                            RES = current residuals (non-dim)
C     OUTPUT VARIABLE:       DRDQ = numeric derivative matrix (1/m)
C
      INCLUDE      PARAM.INC
      INCLUDE      GLBBLK.INC
      INTEGER      BEN_TOR,MBT
      REAL         VEL,AOA,FREQ,QLIT(MAXMODE,3),RES(3*MAXMODE),RGEOM
      REAL         VEL2,AOA2,FREQ2,QLIT2(MAXMODE,3),RES2(3*MAXMODE)
      REAL         QBIG(MAXMODE,3),DRDQ(3*MAXMODE,3*MAXMODE)
C
      MBT = (BEN_TOR-1)*NB + 1
C
C     Loop through each direction of the components of the modal
C     amplitudes, ignoring oscillating components if steady analysis.
C
      MAX = 3
      IF (STEADY) MAX = 1
      DO 30 I1 = 1,NMODES
      DO 30 J1 = 1,MAX
C
C        Skip if looking at components of state vector
C        reserved for angle of attack/velocity and reduced frequency.
C
         IF ((I1.EQ.MBT).AND.(J1.NE.1)) GOTO 30
C
```

234

```
C           Initialize modal amplitude trial vector.
C
            DO 10 I2 = 1,MAXMODE
            DO 10 J2 = 1,3
               QLIT2(I2,J2) = QLIT(I2,J2)
10          CONTINUE
C
C           Increment desired direction of modal amplitude
C           trial vector by 0.1%
C
            QLIT2(I1,J1) = 1.001*QLIT(I1,J1)
            IF (ABS(QLIT2(I1,J1)).LT.1.E-4) QLIT2(I1,J1)=0.^01
C
C           Calculate new residuals from modal amplitude trial vector.
C
            CALL RESIDUAL(VEL,AOA,FREQ,QLIT2,RGEOM,RES2,QBIG)
C
C           Calculate numeric derivatives from modal amplitude trial
C           vector QLIT2 and associated residuals RES2.
C
            K1 = NMODES*(J1-1)+I1
            DO 20 I2 = 1,NMODES
            DO 20 J2 = 1,MAX
               K2 = NMODES*(J2-1)+I2
               DRDQ(K2,K1) = (RES2(K2)-RES(K2))/(QLIT2(I1,J1)-
     &         QLIT(I1,J1))
20          CONTINUE
30        CONTINUE
C
C         If steady, skip angle of attack/velocity and frequency
C         derivatives.
C
          IF (STEADY) GOTO 60
C
C         Increment trial angle of attack/velocity by 0.1% and
C         calculate new residuals.
C
          IF (VLINES) THEN
             AOA2 = 1.001*AOA
             IF (AOA2.EQ.0.) AOA2=0.001
             CALL RESIDUAL(VEL,AOA2,FREQ,QLIT,RGEOM,RES2,QBIG)
          ELSE
             VEL2 = 1.001*VEL
             IF (VEL2.EQ.0.) VEL2=0.001
             CALL RESIDUAL(VEL2,AOA,FREQ,QLIT,RGEOM,RES2,QBIG)
          ENDIF
C
C         Calculate numeric derivatives from trial angle of attack/velocity
C         AOA2/VEL2 and associated residuals RES2.
C
          K1 = NMODES+MBT
          DO 40 I2 = 1,NMODES
          DO 40 J2 = 1,MAX
             K2 = NMODES*(J2-1)+I2
             IF (VLINES) DRDQ(K2,K1)=(RES2(K2)-RES(K2))/(AOA2-AOA)
             IF (.NOT.VLINES) DRDQ(K2,K1)=(RES2(K2)-RES(K2))/
     &       (VEL2**2-VEL**2)
40        CONTINUE
C
C         Increment trial frequency by 0.1% and calculate new residuals.
C
          FREQ2 = 1.001*FREQ
          IF (FREQ2.EQ.0.) FREQ2=0.001
          CALL RESIDUAL(VEL,AOA,FREQ2,QLIT,RGEOM,RES2,QBIG)
```

```
C
C       Calculate numeric derivatives from trial frequency FREQ2 and
C       associated residuals RES2.
C
        K1 = 2*NMODES+MBT
        DO 50 I2 = 1,NMODES
        DO 50 J2 = 1,MAX
           K2 = NMODES*(J2-1)+I2
           DRDQ(K2,K1)=(RES2(K2)-RES(K2))/(FREQ2-FREQ)
50      CONTINUE
C
60      RETURN
        END

C-----FILE: MASS.FOR -----------------------------------------------
C
        SUBROUTINE MASS(LO,HI,MPA)
C
C       Subroutine to calculate components of the flat plate mass matrix
C
        INCLUDE      PARAM.INC
        INCLUDE      GLBBLK.INC
        REAL         LO,HI,MPA,INTGRL
C
        DO 10 I = 1,NMODES
        DO 10 J = I,NMODES
           IF ((I.LE.NB+NT+NC).AND.(J.LE.NB+NT+NC)) THEN
C
C          Calculate out-of-plane mass matrix components.
C
           M(I,J) = MPA*CHORD*LENGTH*INTGRL('X',I,0,J,0,LO,HI)*
     &        INTGRL('Y',I,0,J,0,-.5,+.5)
           M(J,I) = M(I,J)
        ELSEIF ((I.GT.NB+NT+NC).AND.(J.GT.NB+NT+NC)) THEN
C
C          Calculate fore-&-aft mass matrix components.
C
           M(I,J) = MPA*CHORD*LENGTH*INTGRL('X',I,0,J,0,LO,HI)*
     &        INTGRL('Y',I,0,J,0,-.5,+.5)
           M(J,I) = M(I,J)
        ELSE
C
C          Calculate out-of-plane/fore-&-aft coupling
C          mass matrix components.
C
           M(I,J) = 0.
           M(J,I) = 0.
        ENDIF
10      CONTINUE
C
        RETURN
        END

C-----FILE: MODE.FOR -----------------------------------------------
C
        REAL FUNCTION FMODE(DERIV,XY,NUM,INPUT)
C
C       X and Y variation of the five assumed modes. Note that
C       all the x and y coordinates have already been normalized.
C       DERIV indicates what derivative of the mode is given.
C
        INCLUDE      PARAM.INC
        INCLUDE      GLBBLK.INC
        INTEGER      DERIV,NUM,NUMBF
```

236

```fortran
      CHARACTER  XY*1
      REAL       INPUT,RN(5),EPS,ALF
      DATA       RN / 0.596864162695,1.494175614274,
     &           2.500246946168,3.499989319849,
     &           4.500000461516 /
C
      IF (NUM.EQ.0) THEN
         IF (XY.EQ.'X') THEN
            IF (DERIV.EQ.0) THEN
               FMODE = 1./3.*(INPUT-1.)**4+4./3.*(INPUT-1.)+1.
            ELSEIF (DERIV.EQ.1) THEN
               FMODE = 4./3.*(INPUT-1.)**3+4./3.
            ELSEIF (DERIV.EQ.2) THEN
               FMODE = 4.*(INPUT-1.)**2
            ELSEIF (DERIV.EQ.3) THEN
               FMODE = 8.*(INPUT-1.)
            ELSEIF (DERIV.EQ.4) THEN
               FMODE = 8.
            ELSEIF (DERIV.GE.5) THEN
               FMODE = 0.
            ENDIF
         ELSEIF (XY.EQ.'Y') THEN
            IF (DERIV.EQ.0) FMODE = 1.
            IF (DERIV.GT.0) FMODE = 0.
         ENDIF
      ELSEIF ((XY.EQ.'X').AND.((NUM.LE.NB).OR.(NUM.GT.NB+NT+NC))) THEN
C
C        Describe DERIV-th derivative of the spanwise, x-variation
C        of the bending modes or fore-&-aft modes.
C
         NUMBF = NUM
         IF (NUM.GT.NB) NUMBF = NUM - (NB+NT+NC)
         IF (NUMBF.LE.5) EPS = RN(NUMBF)*PI
         IF (NUMBF.GT.5) EPS = (REAL(NUMBF)-.5)*PI
         ALF = (SINH(EPS)-SIN(EPS))/(COSH(EPS)+COS(EPS))
C
         IF (DERIV.EQ.0) THEN
            FMODE = COSH(EPS*INPUT)-ALF*SINH(EPS*INPUT)-
     &         COS(EPS*INPUT)+ALF*SIN(EPS*INPUT)
         ELSEIF ((DERIV.GT.0).AND.(MOD(DERIV,2).EQ.1)) THEN
            FMODE = (EPS**DERIV)*(SINH(EPS*INPUT)-ALF*
     &         COSH(EPS*INPUT)+(SIN(EPS*INPUT)+ALF*
     &         COS(EPS*INPUT))*((-1)**((DERIV+3)/2)))
         ELSEIF ((DERIV.GT.0).AND.(MOD(DERIV,2).EQ.0)) THEN
            FMODE = (EPS**DERIV)*(COSH(EPS*INPUT)-ALF*
     &         SINH(EPS*INPUT)+(COS(EPS*INPUT)-ALF*
     &         SIN(EPS*INPUT))*((-1)**((DERIV+2)/2)))
         ENDIF
      ELSEIF ((XY.EQ.'X').AND.(NUM.GT.NB).AND.(NUM.LE.NB+NT)) THEN
C
C        Describe the DERIV-th derivative of the spanwise, x-variation
C        of the torsional modes.
C
         IT = NUM - NB
         IF (DERIV.EQ.0) THEN
            FMODE = B(IT,1)*COS(G(IT)*INPUT)+B(IT,2)*
     &         SIN(G(IT)*INPUT)+B(IT,3)*COSH(F(IT)*INPUT)+
     &         B(IT,4)*SINH(F(IT)*INPUT)
         ELSEIF ((DERIV.GT.0).AND.(MOD(DERIV,2).EQ.1)) THEN
            FMODE = (G(IT)**DERIV)*(-B(IT,1)*SIN(G(IT)*
     &         INPUT)+B(IT,2)*COS(G(IT)*INPUT))*((-1)**((DERIV+3)/
     &         2))+(F(IT)**DERIV)*(B(IT,3)*SINH(F(IT)*INPUT)+
     &         B(IT,4)*COSH(F(IT)*INPUT))
         ELSEIF ((DERIV.GT.0).AND.(MOD(DERIV,2).EQ.0)) THEN
```

```
                  FMODE = (G(IT)**DERIV)*(-B(IT,1)*COS(G(IT)*INPUT)-
     &            B(IT,2)*SIN(G(IT)*INPUT))*((-1)**((DERIV+2)/2))+
     &            (F(IT)**DERIV)*(B(IT,3)*COSH(F(IT)*INPUT)+B(IT,4)*
     &            SINH(F(IT)*INPUT))
              ENDIF
C
C         IF (DERIV.EQ.0) THEN
C             FMODE = SIN(REAL(2*IT-1)*PI/2.*INPUT)
C         ELSEIF ((DERIV.GT.0).AND.(MOD(DERIV,2).EQ.1)) THEN
C             FMODE = COS(REAL(2*IT-1)*PI/2.*INPUT)*(REAL(2*
C     &            IT-1)*PI/2.)**DERIV*(-1)**((DERIV+3)/2)
C         ELSEIF ((DERIV.GT.0).AND.(MOD(DERIV,2).EQ.0)) THEN
C             FMODE = SIN(REAL(2*IT-1)*PI/2.*INPUT)*(REAL(2*
C     &            IT-1)*PI/2.)**DERIV*(-1)**(DERIV/2)
C         ENDIF
          ELSEIF ((XY.EQ.'X').AND.(NUM.EQ.NB+NT+1).AND.(NC.GE.1)) THEN
C
C         Describe the DERIV-th derivative of the spanwise, x-variation
C         of the 1st chordwise bending mode.
C
              IF (DERIV.EQ.0) FMODE = INPUT*(1.-INPUT)
              IF (DERIV.EQ.1) FMODE = 1.-2.*INPUT
              IF (DERIV.EQ.2) FMODE = -2.
              IF (DERIV.GE.3) FMODE = 0.
          ELSEIF ((XY.EQ.'X').AND.(NUM.EQ.NB+NT+2).AND.(NC.GE.2)) THEN
C
C         Describe the DERIV-th derivative of the spanwise, x-variation
C         of the 2nd chordwise bending mode.
C
              IF (DERIV.EQ.0) FMODE = INPUT**2 - 1.
              IF (DERIV.EQ.1) FMODE = 2.*INPUT
              IF (DERIV.EQ.2) FMODE = 2.
              IF (DERIV.GE.3) FMODE = 0.
          ELSEIF ((XY.EQ.'Y').AND.((NUM.LE.NB).OR.(NUM.GT.NB+NT+NC))) THEN
C
C         Describe the DERIV-th derivative of the chordwise, y-variation
C         of the bending modes or fore-&-aft modes.
C
              IF (DERIV.EQ.0) FMODE = 1.
              IF (DERIV.GE.1) FMODE = 0.
          ELSEIF ((XY.EQ.'Y').AND.(NUM.GT.NB).AND.(NUM.LE.NB+NT)) THEN
C
C         Describe the DERIV-th derivative of the chordwise, y-variation
C         of the torsional modes.
C
              IF (DERIV.EQ.0) FMODE = INPUT
              IF (DERIV.EQ.1) FMODE = 1.
              IF (DERIV.GE.2) FMODE = 0.
          ELSEIF ((XY.EQ.'Y').AND.(NUM.EQ.NB+NT+1).AND.(NC.GE.1)) THEN
C
C         Describe the DERIV-th derivative of the chordwise, y-variation
C         of the 1st chordwise bending mode.
C
              IF (DERIV.EQ.0) FMODE = (4.*INPUT*INPUT - 1./3.)
              IF (DERIV.EQ.1) FMODE = 8.*INPUT
              IF (DERIV.EQ.2) FMODE = 8.
              IF (DERIV.GE.3) FMODE = 0.
          ELSEIF ((XY.EQ.'Y').AND.(NUM.EQ.NB+NT+2).AND.(NC.GE.2)) THEN
C
C         Describe the DERIV-th derivative of the chordwise, y-variation
C         of the 2nd chordwise bending mode.
C
              IF (DERIV.EQ.0) FMODE = (2.*INPUT)**2 - 1
              IF (DERIV.EQ.1) FMODE = 8.*INPUT
```

```
          IF (DERIV.EQ.2) FMODE = 8.
          IF (DERIV.GE.3) FMODE = 0.
       ELSE
          FMODE = 0.
       ENDIF
C
       RETURN
       END

C-----FILE: QBIG.FOR ----------------------------------------------------
C
       SUBROUTINE MODAL_FORCE(VEL,AOA,FREQ,QLIT,QBIG)
C
C      Subroutine to calculate the oscillating components of the modal
C      forces.
C
C      INPUT VARIABLES:   VEL = velocity (m/s)
C                         AOA = root angle of attack (rad)
C                         FREQ = reduced frequency (non-dim)
C                         QLIT = modal amplitudes (m)
C      OUTPUT VARIABLE:   QBIG = modal forces (N)
C
       INCLUDE     PARAM.INC
       INCLUDE     GLBBLK.INC
       REAL        VEL,AOA,FREQ,QLIT(MAXMODE,3),QBIG(MAXMODE,3)
       REAL        THETA(3),HBAR(3),VBAR(3)
       REAL        DCL(0:2),CL(5),CM(5),CD(5)
       INCLUDE     GAUSS.INC
C
C      THETA:      Oscillating components of real angle of attack (rad)
C      HBAR:       Oscillating components of 1/4-chord out-of-plane
C                  deflection, non-dimensionalized with respect to
C                  the half-chord
C      VBAR:       Oscillating components of 1/4-chord fore-&-aft
C                  deflection, non-dimensionalized with respect to
C                  the half-chord
C      DCL:        Oscillating components of the static deviation from
C                  the linear lift curve (non-dim)
C      CL:         Oscillating components of the lift coeff (non-dim)
C      CM:         Oscillating components of the moment coeff (non-dim)
C      CD:         Oscillating components of the drag coeff (non-dim)
C
C      Initialize the modal forces to zero value.
C
       DO 10 I = 1,MAXMODE
       DO 10 J = 1,3
          QBIG(I,J) = 0.
10     CONTINUE
C
C      Loop through Gauss integration points along the span.
C
       DO 60 IGNUM = 1,GPOINTS
C
C         Calculate the non-dimensional 1/4-chord deflection,
C         angle-of-attack, and fore-&-aft sinusoidal coefficients
C         at the Gauss point spanwise location.
C
          XBAR = (GP(IGNUM)+1.)/2.
          DO 30 I = 1,3
C
C            Add contributions to out-of-plane deflection
C            and to torsional twist.
C
             HBAR(I) = 0.
```

239

```fortran
                  THETA(I) = 0.
                  DO 20 J = 1,NB+NT+NC
                     HBAR(I) = HBAR(I) + QLIT(J,I)/(CHORD/2.)*
     &               FMODE(0,'X',J,XBAR)*FMODE(0,'Y',J,+.25)
                     THETA(I) = THETA(I) + QLIT(J,I)/CHORD*
     &               FMODE(0,'X',J,XBAR)*FMODE(1,'Y',J,+.25)
   20             CONTINUE
                  IF (LATAN) THETA(I) = ATAN(THETA(I))
                  IF (I.EQ.1) THETA(I)=THETA(I)+AOA
C
C                 Add contributions to fore-&-aft deflection.
C
                  VBAR(I) = 0.
                  IF (NF.GT.0) THEN
                  DO 25 J = NB+NT+NC+1,NB+NT+NC+NF
                     VBAR(I) = VBAR(I) + QLIT(J,I)/(CHORD/2.)*
     &               FMODE(0,'X',J,XBAR)*FMODE(0,'Y',J,+.25)
   25             CONTINUE
                  ENDIF
   30          CONTINUE
C
C              Calculate the lift/moment coefficient sinusoidal coefficients.
C
               CALL AEROF('L',THETA,HBAR,VEL,FREQ,LAEROF,DCL,CL)
               CALL AEROF('M',THETA,HBAR,VEL,FREQ,LAEROF,DCL,CM)
C
C              Calculate the profile-drag coefficient contribution using
C              a 3rd-order polynomial fit.
C
               CD(1) = 4.923*ABS(THETA(1))**3 + .1472*THETA(1)**2 + .042*
     &         ABS(THETA(1)) + .014
               CD(2) = 0.
               CD(3) = 0.
C
C              Calculate the induced-drag coefficient contribution.
C
               CD(1) = CD(1) + CL(1)**2/PI/(2.*LENGTH/CHORD)
C
C              Incorporate spanwise correction for the force dropoff.
C
               DUMMY = 1.
               IF (CORREC) DUMMY = SC(XBAR)
C
C              Add contributions from the lift, moment, and drag at the
C              current Gauss point spanwise location to the modal force.
C
               DO 50 I = 1,NMODES
               DO 50 J = 1,3
                  IF (I.LE.NB+NT+NC) THEN
                     QBIG(I,J) = QBIG(I,J)+GW(IGNUM)/2.*(.5*RHOA*
     &               VEL**2*CHORD*LENGTH)*((CL(J)*COS(AOA)+CD(J)*
     &               SIN(AOA))*FMODE(0,'Y',I,+.25)+CM(J)*
     &               FMODE(1,'Y',I,+.25))*FMODE(0,'X',I,XBAR)*DUMMY
                  ELSE
                     QBIG(I,J) = QBIG(I,J)+GW(IGNUM)/2.*(.5*RHOA*
     &               VEL**2*CHORD*LENGTH)*(-CL(J)*SIN(AOA)+CD(J)*
     &               COS(AOA))*FMODE(0,'Y',I,+.25)*FMODE(0,'X',I,XBAR)*
     &               DUMMY
                  ENDIF
   50          CONTINUE
   60       CONTINUE
C
            RETURN
            END
```

240

```
C-----FILE : RESIDUAL.FOR ---------------------------------------------
C
      SUBROUTINE RESIDUAL(VEL,AOA,FREQ,QLIT,RGEOM,RES,QBIG)
C
C     Subroutine to calculate the residuals used in the Newton-Raphson
C     solver.
C
C     INPUT  VARIABLES:  VEL = velocity (m/s)
C                        AOA = root angle of attack (rad)
C                        FREQ = reduced frequency (non-dim)
C                        QLIT = modal amplitudes (m)
C     OUTPUT VARIABLES:  RES = residuals, non-dimensionalized by
C                              1/2*rho*(V**2)*area
C                        QBIG = modal forces, non-dimensionalized by
C                              1/2*rho*(V**2)*area
C
      INCLUDE     PARAM.INC
      INCLUDE     GLBBLK.INC
      REAL        VEL,AOA,FREQ,QLIT(MAXMODE,3),RGEOM
      REAL        RES(3*MAXMODE),QBIG(MAXMODE,3)
C
C     Calculate the modal forces QBIG using subroutine MODAL_FORCE,
C     which are functions of the velocity VEL, the root angle of attack
C     AOA, the reduced frequency FREQ, and the modal amplitudes QLIT.
C
      CALL MODAL_FORCE(VEL,AOA,FREQ,QLIT,QBIG)
C
      DO 30 I1 = 1,MAXMODE
      DO 30 I2 = I1,MAXMODE
         KDUM(I1,I2) = K(I1,I2)
C
C        Add corrections to stiffness matrix for geometric
C        nonlinearities.
C
         IF (LGEOM) THEN
            IF ((I1.GT.NB).AND.(I1.LE.NB+NT).AND.
     &         (I2.GT.NB).AND.(I2.LE.NB+NT)) THEN
               DO 10 I3 = 1,NB
               DO 10 I4 = 1,NB
                  KDUM(I1,I2) = KDUM(I1,I2) + RGEOM**2*
     &               R(I3,I4,I1-NB,I2-NB)*QLIT(I3,1)*QLIT(I4,1)
10             CONTINUE
            ELSEIF ((I1.GT.NB).AND.(I1.LE.NB+NT).AND.
     &         (I2.GT.NB+NT+NC).AND.(I2.LE.NB+NT+NC+NF)) THEN
               DO 20 I3 = 1,NB
                  KDUM(I1,I2) = KDUM(I1,I2) + RGEOM*
     &               H(I3,I1-NB,I2-NB-NT-NC)*QLIT(I3,1)
20             CONTINUE
            ENDIF
         ENDIF
C
         KDUM(I2,I1) = KDUM(I1,I2)
30    CONTINUE
C
      KTT0 = KDUM(NB+1,NB+1)
C
C     Calculate the residuals by including the contributions
C     of the mass and stiffness matrices with the modal forces.
C
      DO 60 J = 1,3
C
C        Add correction to nonlinear stiffness matrix for cubic
C        stiffening.
```

241

```
C
          IF (J.EQ.1) KDUM(NB+1,NB+1) = KTT0 + KTTCUBE*
     &     (QLIT(NB+1,1)**2+1.5*QLIT(NB+1,2)**2+1.5*
     &     QLIT(NB+1,3)**2)
          IF (J.NE.1) KDUM(NB+1,NB+1) = KTT0 + KTTCUBE*
     &     (3.*QLIT(NB+1,1)**2+.75*QLIT(NB+1,2)**2+
     &     .75*QLIT(NB+1,3)**2)
C
          DO 50 I = 1,NMODES
             II = NMODES*(J-1)+I
             IF ((STEADY).AND.(J.NE.1)) THEN
                RES(II) = 0.
             ELSE
                RES(II) = -QBIG(I,J)
                DO 40 L = 1,NMODES
                   RES(II) = RES(II)+KDUM(I,L)*QLIT(L,J)
                   OMEGA = FREQ*VEL/(CHORD/2.)
                   IF (J.NE.1) RES(II)=RES(II)-OMEGA**2*
     &                M(I,L)*QLIT(L,J)
40              CONTINUE
             ENDIF
             RES(II) = RES(II)/(.5*RHOA*VEL**2*CHORD*LENGTH)
             QBIG(I,J) = QBIG(I,J)/(.5*RHOA*VEL**2*CHORD*LENGTH)
50        CONTINUE
60     CONTINUE
C
       RETURN
       END

C-----FILE: SETMODE.FOR ----------------------------------------------
C
       SUBROUTINE SETMODE(D11,D66,CHORD,LENGTH,NT,NTMAX,KT,G,F,B,ERR)
C
C      Subroutine to compute the coefficients G(I), F(I), and B(I,4) of
C      the torsional mode shapes for a laminate, given its aspect ratio
C      and its bending and torsion stiffness properties, D11 and D66,
C      based on Crawley & Dugundji, "Frequency Determination and Non-
C      Dimensionalization for Composite Cantilever Plates," Journal
C      of Sound and Vibration, Vol. 72, No. 1, 1980, pp. 1-10.
C
       INTEGER  NT,NTMAX
       REAL     D11,D66,CHORD,LENGTH
       REAL     KT(NTMAX),G(NTMAX),F(NTMAX),B(NTMAX,4),ERR(NTMAX)
       REAL     PI,BETA,A(4,4),BDUM(4),XDUM(4),R0,R1,DELTA
       LOGICAL  CONVERGED
C
C      Initialize warping stiffness influence coefficient.
C
       PI = ABS(ACOS(-1.))
       BETA = D11*CHORD**2/(48.*D66*LENGTH**2)
C
C      Loop thru torsion modes, calculating appropriate coeffs.
C
       DO 30 I = 1,NT
C
C         Initialize guesses at non-dimensional natural frequency
C         (start a little above the value for zero warping stiffness)
C         and sin/cos coefficient - derived by inverting eqn [17]
C         from Crawley & Dugundji.
C
          IF (I.EQ.1) KT(I) = PI/2.
          IF (I.NE.1) KT(I) = KT(I-1)*REAL(2*I-1)/
     &     REAL(2*I-3) * 1.20
          G(I) = SQRT((SQRT(1.+4.*BETA*KT(I)**2)-1.)/(2.*BETA))
```

242

```fortran
          F(I) = SQRT(G(I)**2+1./BETA)
C
          CONVERGED = .FALSE.
          DO WHILE (.NOT.CONVERGED)
C
C           Calculate residual, R0 — i.e. determinant of 4x4 matrix in
C           eqn [18] of Crawley & Dugundji — that is to be driven
C           to zero (i.e. to make matrix singular, i.e. a natural
C           mode). Also calculate derivative of residual w.r.t. G(I),
C           R1. Both calculations done by Mathematica™.
C
            R0 = cosh(F(I))**2*F(I)**2*G(I)**2 - F(I)**2*G(I)**2*
     &       sinh(F(I))**2 + cosh(F(I))*F(I)**4*Cos(G(I)) +
     &       cosh(F(I))*G(I)**4*Cos(G(I)) + F(I)**2*G(I)**2*
     &       Cos(G(I))**2 + F(I)**3*G(I)*sinh(F(I))*Sin(G(I)) -
     &       F(I)*G(I)**3*sinh(F(I))*Sin(G(I)) + F(I)**2*
     &       G(I)**2*Sin(G(I))**2
            R1 = 2*cosh(F(I))**2*F(I)**2*G(I) + 2*cosh(F(I))**2*
     &       G(I)**3 - 2*F(I)**2*G(I)*sinh(F(I))**2 - 2*G(I)**3*
     &       sinh(F(I))**2 + 4*cosh(F(I))*F(I)**2*G(I)*Cos(G(I)) +
     &       4*cosh(F(I))*G(I)**3*Cos(G(I)) + 2*F(I)**3*G(I)*
     &       sinh(F(I))*Cos(G(I)) - F(I)*G(I)**3*sinh(F(I))*
     &       Cos(G(I)) + (G(I)**5*sinh(F(I))*Cos(G(I)))/F(I) +
     &       2*F(I)**2*G(I)*Cos(G(I))**2 + 2*G(I)**3*
     &       Cos(G(I))**2 - cosh(F(I))*F(I)**4*Sin(G(I)) +
     &       cosh(F(I))*F(I)**2*G(I)**2*Sin(G(I)) - 2*cosh(F(I))*
     &       G(I)**4*Sin(G(I)) + F(I)**3*sinh(F(I))*Sin(G(I)) -
     &       (G(I)**4*sinh(F(I))*Sin(G(I)))/F(I) + 2*F(I)**2*G(I)*
     &       Sin(G(I))**2 + 2*G(I)**3*Sin(G(I))**2
C
C           Apply Newton's Method and test relative convergence.
C
            DELTA = R0/R1
            G(I) = G(I) - DELTA
            F(I) = SQRT(G(I)**2+1./BETA)
            IF ((ABS(DELTA/G(I)).LT.(1.E-7)).AND.(ABS(R0/
     &       (F(I)**4*COSH(F(I)))).LT.(1.E-6))) CONVERGED=.TRUE.
          END DO
C
C         Calculate converged value of non-dimensional natural frequency.
C
          KT(I) = G(I)*SQRT(1.+BETA*G(I)**2)
C
C         Set up matrix equation to solve for the mode shape, i.e.
C         eqn [18] from Crawley & Dugundji with the 4th row of the
C         matrix equation converted to reflect a non-dimensional tip
C         deflection of 1.
C
          A(1,1) = 1.
          A(1,2) = 0.
          A(1,3) = 1.
          A(1,4) = 0.
          A(2,1) = 0.
          A(2,2) = G(I)
          A(2,3) = 0.
          A(2,4) = F(I)
          A(3,1) = -G(I)**2*COS(G(I))
          A(3,2) = -G(I)**2*SIN(G(I))
          A(3,3) =  F(I)**2*COSH(F(I))
          A(3,4) =  F(I)**2*SINH(F(I))
          A(4,1) = COS(G(I))
          A(4,2) = SIN(G(I))
          A(4,3) = COSH(F(I))
          A(4,4) = SINH(F(I))
```

243

```fortran
                BDUM(1) = 0.
                BDUM(2) = 0.
                BDUM(3) = 0.
                BDUM(4) = 1.
c
c               1st row converted to ND tip deflection of 1.
c
c               A(1,1) = COS(G(I))
c               A(1,2) = SIN(G(I))
c               A(1,3) = COSH(F(I))
c               A(1,4) = SINH(F(I))
c               A(4,1) =   F(I)**2*G(I)*SIN(G(I))
c               A(4,2) = -F(I)**2*G(I)*COS(G(I))
c               A(4,3) =   F(I)*G(I)**2*SINH(F(I))
c               A(4,4) =   F(I)*G(I)**2*COSH(F(I))
c               BDUM(1) = 1.
c               BDUM(4) = 0.
c
                CALL SOLVE(A,BDUM,XDUM,4,1,4)
                DO 20 J = 1,4
                    B(I,J) = XDUM(J)
20              CONTINUE
c
c               Set the error function equal to the unresolved 4th equation
c               of eqn [18] from Crawley & Dugundji.
c
                ERR(I) = B(I,1)*F(I)**2*G(I)*SIN(G(I)) -
     &            B(I,2)*F(I)**2*G(I)*COS(G(I)) + B(I,3)*F(I)*
     &            G(I)**2*SINH(F(I)) + B(I,4)*F(I)*G(I)**2*COSH(F(I))
c
c               Unresolved 1st equation.
c
c               ERR(I) = B(I,1) + B(I,3)
c
30          CONTINUE
c
            RETURN
            END

C-----FILE: SETUP.FOR -----------------------------------------------
C
        REAL FUNCTION INTGRL(XY,MODE1,DERIV1,MODE2,DERIV2,LO,HI)
C
C       Subroutine to integrate numerically the DERIV1-th derivative
C       of the XY-variation of MODE1 with the DERIV2-th derivative of
C       the XY-variation of MODE2 over the normalized interval [LO,HI],
C       using a GPOINTS Gaussian quadrature scheme ("Handbook of
C       Mathematical Functions,"Abramowitz and Stegun (eds.), Table
C       25.4, p.916).
C
        INTEGER     MODE1,MODE2,DERIV1,DERIV2
        REAL        LO,HI,POINT
        CHARACTER   XY*1
        INCLUDE     GAUSS.INC
C
        INTGRL = 0.
        DO 10 I = 1,GPOINTS
            POINT = (GP(I)*(HI-LO)+HI+LO)/2.
            INTGRL = INTGRL + GW(I)*FMODE(DERIV1,XY,MODE1,POINT)*
     &        FMODE(DERIV2,XY,MODE2,POINT)*(HI-LO)/2.
10      CONTINUE
C
        RETURN
        END
```

244

```fortran
C
C========================================================================
C
      REAL FUNCTION SC_INT(MODE1,MODE2)
C
C     Subroutine to integrate numerically the X-variation of MODE1 with
C     the X-variation of MODE2 over the normalized interval [0,1],
C     with spanwise correction SC(X) included in the integral,
C     using a GPOINTS Gaussian quadrature scheme ("Handbook of
C     Mathematical Functions,"Abramowitz and Stegun (eds.), Table
C     25.4, p.916).
C
      INTEGER      MODE1,MODE2
      REAL         POINT
      INCLUDE      GAUSS.INC
C
      SC_INT = 0.
      DO 10 I = 1,GPOINTS
         POINT = (GP(I)+1.)/2.
         SC_INT = SC_INT + GW(I)*FMODE(0,'X',MODE1,POINT)*
     &     FMODE(0,'X',MODE2,POINT)/2.*SC(POINT)
10    CONTINUE
C
      RETURN
      END

C-----FILE: STATIC.FOR ------------------------------------------------
C
      SUBROUTINE STATIC(LAYUP,LSTRUC,TRATIO,IERR)
C
C     Subroutine to calculate mass and stiffness matrices and to run
C     static deflection and free vibration analyses.
C
      INCLUDE      PARAM.INC
      INCLUDE      GLBBLK.INC
      REAL         TRATIO,STYLO,ELFOAM,ETFOAM,NULTFOAM,GLTFOAM
      REAL         RHOFOAM,THETA(MAXPLIES)
      REAL         ZU(MAXPLIES),ZL(MAXPLIES)
      REAL         EL(MAXPLIES),ET(MAXPLIES),NULT(MAXPLIES)
      REAL         GLT(MAXPLIES),RHO(MAXPLIES),QU11(MAXPLIES)
      REAL         QU12(MAXPLIES),QU22(MAXPLIES),QU66(MAXPLIES)
      REAL         QT(3,3,MAXPLIES),ERR(NTMAX)
      REAL         A(3,3),D(3,3),INTGRL
      REAL         MPA,DFOAM(3,3),MFOAM(MAXMODE,MAXMODE)
      REAL         KFOAM(MAXMODE,MAXMODE),QLIT(MAXMODE)
      REAL         DRDQ(MAXMODE,MAXMODE),DQ(MAXMODE),RES(MAXMODE)
      INTEGER      IERR,NPLIES
      LOGICAL      LSTRUC,EQUAL,CONVERGED
      CHARACTER    LAYUP*25,ANSWER*1,FILENAME*25
      INCLUDE      GAUSS.INC
C
C     Loop through flat plate (I1=1) and NACA wing (I1=2) analyses.
C
      DO 440 I1 = 1,2
C
      IF (I1.EQ.1) THEN
C
C     If first loop, then calculate flat plate mass and stiffness
C     properties. Begin by reading in foam and ply data from input file
C     and creating unidirectional Q-values [QU] and rotated Q-values,
C     ie. Q-theta [QT] for each ply.
C
      FILENAME = TRIM(LAYUP)//'.DAT '
      OPEN(2,FILE=TRIM(FILENAME),STATUS='OLD',FORM='FORMATTED',
```

```fortran
     &    IOSTAT=IERR)
      IF (IERR.NE.0) GOTO 999
      WRITE(*,'(/2A)') 'Layup data taken from file ',TRIM(FILENAME)
      READ(2,*) CHORD,LENGTH,NPLIES,LAMBDA,KTTCUBE
      READ(2,*) TRATIO,STYLO,ELFOAM,ETFOAM,NULTFOAM,GLTFOAM,RHOFOAM
      DO 10 I = 1,NPLIES
          READ(2,*) THETA(I),ZU(I),ZL(I),EL(I),ET(I),NULT(I),
     &        GLT(I),RHO(I)
          THETA(I) = THETA(I)*PI/180.
          CALL QUCON(EL(I),ET(I),NULT(I),GLT(I),QU11(I),QU12(I),
     &        QU22(I),QU66(I))
          CALL QTCON(I,THETA(I),QU11(I),QU12(I),QU22(I),QU66(I),QT)
10    CONTINUE
      CLOSE(2)
C
      IOUT = 0
      IF (LSTRUC) THEN
C
C         Output wing properties.
C
20        WRITE(*,'(A,$)') ' Output engineering properties [2] '//
     &        'to file or [9] to screen ? '
          READ(*,*,ERR=20) IOUT
          IF ((IOUT.NE.2).AND.(IOUT.NE.9)) GOTO 20
          IF (IOUT.EQ.2) THEN
              FILENAME = TRIM(LAYUP)//'PNAS.OUT '
              IF (TRATIO.GT.0.) FILENAME = TRIM(LAYUP)//'WNAS.OUT '
              OPEN(UNIT=IOUT,FILE=TRIM(FILENAME),STATUS='NEW',
     &            FORM='FORMATTED')
              WRITE(*,*) ' Calculated wing properties being sent to ',
     &            TRIM(FILENAME)
              WRITE(IOUT,'(A,4I3)') 'NB, NT, NC, NF =',NB,NT,NC,NF
          ENDIF
      ENDIF
C
      IF (IOUT.NE.0) THEN
C
C         Output basic overall properties.
C
          WRITE(IOUT,'(/1X,A,F5.4,A,F4.1,A)') '      Chord length = ',
     &        CHORD,' m = ',(CHORD*12./.3048),' in'
          WRITE(IOUT,'(1X,A,F5.4,A,F4.1,A)') '        Half span = ',
     &        LENGTH,' m = ',(LENGTH*12./.3048),' in'
          WRITE(IOUT,'(1X,A,I2)') ' Number of plies = ',NPLIES
          WRITE(IOUT,'(A,F5.1,A)') '   Sweepback angle =',
     &        LAMBDA,' deg'
          WRITE(IOUT,'(A,F6.3,A)') '       Air density = ',
     &        RHOA,' kg/m**3'
C
C         Output ply layup in degrees.
C
          WRITE(IOUT,'(A,$)') '             Layup = ['
          DO 30 I = 1,NPLIES
              ANG = THETA(I)*180./PI
              IF (I.NE.1) WRITE(IOUT,'(A,$)') '/'
              IF (ANG.EQ.0.) WRITE(IOUT,'(A,$)') '0'
              IF (ANG.NE.0.) WRITE(IOUT,'(SP,I3,$)') NINT(ANG)
30        CONTINUE
          WRITE(IOUT,'(A,$)') ']'
C
C         Determine if all the plies are of the same material.
C
          EQUAL = .TRUE.
          DO 40 I = 1,NPLIES
```

246

```fortran
                 IF (EL(I).NE.EL(1)) EQUAL=.FALSE.
                 IF (ET(I).NE.ET(1)) EQUAL=.FALSE.
                 IF (GLT(I).NE.GLT(1)) EQUAL=.FALSE.
                 IF (NULT(I).NE.NULT(1)) EQUAL=.FALSE.
                 IF (RHO(I).NE.RHO(1)) EQUAL=.FALSE.
40          CONTINUE
            IF (EQUAL) THEN
                 WRITE(IOUT,'(/1X,A,F4.1,A)') '            GR/EP EL  = ',
     &           EL(1)/1.E9,' GPa'
                 WRITE(IOUT,'(1X,A,F4.1,A)') '            GR/EP ET  = ',
     &           ET(1)/1.E9,' GPa'
                 WRITE(IOUT,'(1X,A,F4.1,A)') '            GR/EP GLT = ',
     &           GLT(1)/1.E9,' GPa'
                 WRITE(IOUT,'(1X,A,F5.2)') 'GR/EP Poisson ratio = ',
     &           NULT(1)
                 WRITE(IOUT,'(1X,A,F6.1,A)') '         GR/EP density = ',
     &           RHO(1),' kg/m**3'
            ENDIF
       ENDIF
C
C       Calculate flat plate A and D matrices (ie. ignoring foam
C       properties).
C
       CALL BEND(NPLIES,ZU,ZL,QT,A,D)
       IF (IOUT.NE.0) THEN
            WRITE(IOUT,'(/A)') ' Flat Plate A-matrix [N/m] : '
            DO 50 I = 1,3
                 WRITE(IOUT,'(10X,3(1PE10.2))') (A(I,J),J=1,3)
50          CONTINUE
            WRITE(IOUT,'(/A)') ' Flat Plate D-matrix [Nm] : '
            DO 60 I = 1,3
                 WRITE(IOUT,'(10X,3F7.4)') (D(I,J),J=1,3)
60          CONTINUE
       ENDIF
C
C       Use the D-matrix properties to set up the natural torsional mode
C       shapes.
C
       CALL SETMODE(D(1,1),D(3,3),CHORD,LENGTH,NT,NTMAX,KT,G,F,B,ERR)
       IF (IOUT.NE.0) THEN
            WRITE(IOUT,'(/1X,A,F8.7)') 'Beta = ',D(1,1)*CHORD**2/
     &      (48.*D(3,3)*LENGTH**2)
            DO 75 I = 1,NT
                 WRITE(IOUT,'(/1X,3(A,I1,A,F7.4))')
     &           'K',I,'T = ',KT(I),';   g',I,' = ',
     &           G(I),';   f',I,' = ',F(I)
72               FORMAT (' B(',I1,') = [',SP,4(1X,F7.5),'] ;  B',
     &           SS,I1,'4-B',I1,'1 =',1PE9.2)
                 WRITE(IOUT,72) I,(B(I,J),J=1,4),I,I,(B(I,4)-B(I,1))
                 WRITE(IOUT,'(A,I1,A,1PE9.2)') ' Err(',I,') =  ',ERR(I)
75          CONTINUE
       ENDIF
C
C       Calculate flat plate mass per unit area thru the thickness
C       (i.e. in the out-of-plane Z-direction).
C
       MPA = 0.
       DO 70 I = 1,NPLIES
            MPA = MPA+(ZU(I)-ZL(I))*RHO(I)
70     CONTINUE
C
C       Calculate flat plate mass matrix.
C
       CALL MASS(0.,1.,MPA)
```

247

```
C
C       Calculate flat plate stiffness matrix.
C
        CALL STIFF(A,D,0.,1.)
        KTT0 = K(NB+1,NB+1)
        KTTCUBE = K(NB+1,NB+1)*KTTCUBE
C
        IF (IOUT.NE.0) THEN
          WRITE(IOUT,'(/A,F5.3,A/)') 'Flat Plate Mass '//
     &      'per unit Area = ',MPA,' kg/m**2'
          WRITE(IOUT,*) 'Flat Plate Mass matrix [kg] : '
          DO 80 I = 1,NMODES
            WRITE(IOUT,'(10X,32(1PE10.2))') (M(I,J),J=1,NMODES)
80        CONTINUE
          WRITE(IOUT,'(/A,1PE8.2,A/)') 'Cubic stiffening = ',
     &      KTTCUBE,' 1/m**2'
          WRITE(IOUT,*) 'Flat Plate Stiffness matrix [N/m] : '
          DO 90 I = 1,NMODES
            WRITE(IOUT,'(10X,32(1PE10.2))') (K(I,J),J=1,NMODES)
90        CONTINUE
        ENDIF
C
C       Initialize sums for numerical integrals used for
C       nonlinear, geometric stiffness corrections.
C
        DO 100 J1 = 0,NBMAX
        DO 100 J2 = 0,NBMAX
        DO 100 J3 = 1,NTMAX
        DO 100 J4 = 1,NTMAX
          R(J1,J2,J3,J4) = 0.
100     CONTINUE
C
        DO 110 J1 = 0,NBMAX
        DO 110 J2 = 1,NTMAX
        DO 110 J3 = 1,NFMAX
          H(J1,J2,J3) = 0.
110     CONTINUE
C
C       Calculate nonlinear torsion-torsion coupling integrals
C       by Gauss numerical integration.
C
        IF (IOUT.NE.0) WRITE(IOUT,*) ' '
        DO 140 J1 = 0,NB
        DO 140 J2 = J1,NB
        DO 140 J3 = 1,NT
        DO 140 J4 = J3,NT
          DO 120 II = 1,GPOINTS
            XBAR = (GP(II)+1.)/2.
            R(J1,J2,J3,J4) = R(J1,J2,J3,J4) + GW(II)/2. *
     &        FMODE(2,'X',J1,XBAR)*FMODE(2,'X',J2,XBAR)*
     &        FMODE(0,'X',NB+J3,XBAR)*FMODE(0,'X',NB+J4,XBAR) *
     &        (A(1,1)*CHORD**2/12.-D(1,1))/(CHORD*LENGTH**3)
120       CONTINUE
          R(J2,J1,J3,J4) = R(J1,J2,J3,J4)
          R(J1,J2,J4,J3) = R(J1,J2,J3,J4)
          R(J2,J1,J4,J3) = R(J1,J2,J3,J4)
130       FORMAT ('R(',2(I1,'B,'),I1,'T,',I1,'T) =',1PE10.3)
          IF (IOUT.NE.0) WRITE(IOUT,130) J1,J2,J3,J4,R(J1,J2,J3,J4)
140     CONTINUE
C
C       Calculate nonlinear fore/aft-torsion cross-coupling integrals
C       by Gauss numerical integration.
C
        IF (NF.GT.0) THEN
```

248

```fortran
      IF (IOUT.NE.0) WRITE(IOUT,*) ' '
      DO 170 J1 = 0,NB
      DO 170 J2 = 1,NT
      DO 170 J3 = 1,NF
          DO 150 II = 1,GPOINTS
              XBAR = (GP(II)+1.)/2.
              H(J1,J2,J3) = H(J1,J2,J3) + G''(II)/2. *
     &         FMODE(2,'X',J1,XBAR)*FMODE(0,'X',NB+J2,XBAR)*
     &         FMODE(2,'X',NB+NT+NC+J3,XBAR) *
     &         (A(1,1)*CHORD**2/12.-D(1,1))/LENGTH**3
150       CONTINUE
160       FORMAT('H(',I1,'B,',I1,'T,',I1,'F) =',1PE10.3)
          IF (IOUT.NE.0) WRITE(IOUT,160) J1,J2,J3,H(J1,J2,J3)
170   CONTINUE
      ENDIF
C
      ELSEIF ((I1.EQ.2).AND.(TRATIO.GT.0.)) THEN
C
C     If second loop, calculate and add styrofoam properties to
C     the mass and stiffness matrices. Begin by outputing foam
C     properties if the foam thickness is non-zero.
C
      IF (IOUT.NE.0) THEN
          WRITE(IOUT,'(/11X,A,F4.1,A)') 'Styrofoam EL  = ',
     &      ELFOAM/1.E6,' MPa'
          WRITE(IOUT,'(11X,A,F4.1,A)') 'Styrofoam ET  = ',
     &      ETFOAM/1.E6,' MPa'
          WRITE(IOUT,'(11X,A,F4.1,A)') 'Styrofoam GLT = ',
     &      GLTFOAM/1.E6,' MPa'
          WRITE(IOUT,'(1X,A,F4.2)') 'Styrofoam Poisson ratio = ',
     &      NULTFOAM
          WRITE(IOUT,'(7X,A,F4.1,A)') 'Styrofoam density = ',
     &      RHOFOAM,' kg/m**3'
          WRITE(IOUT,'(4X,A,F4.3)') 'Styrofoam NACA ratio = ',
     &      TRATIO
      ENDIF
C
C     Calculate the contribution to the mass matrix for the
C     styrofoam bending modes. Note that the styrofoam does
C     not cover the first 2'' (STYLO=1/6) of the span of the airfoil.
C
      DO 180 I = 1,NB
      DO 180 J = I,NB
          MFOAM(I,J) = (0.685*TRATIO*CHORD**2-(ZU(1)-ZL(NPLIES))*
     &      CHORD)*LENGTH*INTGRL('X',I,0,J,0,STYLO,1.)*RHOFOAM
          MFOAM(J,I) = MFOAM(I,J)
180   CONTINUE
C
C     Calculate the contribution to the mass matrix for the
C     styrofoam torsion modes.
C
      DO 190 I = NB+1,NB+NT
      DO 190 J = I,NB+NT
          MFOAM(I,J) = (0.506*TRATIO*CHORD**2-(ZU(1)-ZL(NPLIES))*
     &      CHORD)/12.*LENGTH*INTGRL('X',I,0,J,0,STYLO,1.)*RHOFOAM
          MFOAM(J,I) = MFOAM(I,J)
190   CONTINUE
C
C     Calculate the contribution to the mass matrix for the
C     styrofoam bending-torsion coupling.
C
      DO 200 I = 1,NB
      DO 200 J = NB+1,NB+NT
          MFOAM(I,J) = -0.0545*RHOFOAM*TRATIO*CHORD**2*LENGTH*
```

```fortran
     &      INTGRL('X',I,0,J,0,STYLO,1.)
            MFOAM(J,I) = MFOAM(I,J)
200   CONTINUE
C
C     Calculate the contribution to the mass matrix for the
C     styrofoam chordwise bending modes.
C
      IF (NC.GT.0) THEN
      DO 210 I = 1,NB+NT+NC
      DO 210 J = MAX0(I,NB+NT+1),NB+NT+NC
         MFOAM(I,J) = (0.685*TRATIO*CHORD-(ZU(1)-
     &      ZL(NPLIES)))*CHORD*LENGTH*RHOFOAM*
     &      INTGRL('X',I,0,J,0,STYLO,1.)*
     &      INTGRL('Y',I,0,J,0,-.5,+ 5)
         MFOAM(J,I) = MFOAM(I,J)
210   CONTINUE
      ENDIF
C
C     Calculate the contribution to the mass matrix for the
C     styrofoam fore-&-aft modes.
C
      IF (NF.GT.0) THEN
      DO 220 I = 1,NB+NT+NC+NF
      DO 220 J = MAX0(I,NB+NT+NC+1),NB+NT+NC+NF
         IF (I.LE.NB+NT+NC) THEN
            MFOAM(I,J) = 0.
         ELSE
            MFOAM(I,J) = (0.685*TRATIO*CHORD**2-
     &         (ZU(1)-ZL(NPLIES))*CHORD)*LENGTH*
     &         INTGRL('X',I,0,J,0,STYLO,1.)*RHOFOAM
         ENDIF
         MFOAM(J,I) = MFOAM(I,J)
220   CONTINUE
      ENDIF
C
C     Calculate unidirectional Q-values for the foam. Because the
C     foam is isotropic, these are the same as the rotated Q-values.
C
      CALL QUCON(ELFOAM,ETFOAM,NULTFOAM,GLTFOAM,QT11FOAM,
     & QT12FOAM,QT22FOAM,QT66FOAM)
C
C     Calculate the contribution to the stiffness matrix from
C     the styrofoam bending modes. Note that the styrofoam does
C     not cover the first 2" (STYLO=1/6) of the span of the airfoil.
C
      DO 230 I = 1,NB
      DO 230 J = I,NB
         KFOAM(I,J) = 2.*QT11FOAM*((0.779*TRATIO*CHORD/2.)**3-
     &      ((ZU(1)-ZL(NPLIES))/2.)**3)/3.*CHORD*LENGTH*
     &      INTGRL('X',I,2,J,2,STYLO,1.)/LENGTH**4
         KFOAM(J,I) = KFOAM(I,J)
230   CONTINUE
C
C     Calculate the contribution to the stiffness matrix from
C     the styrofoam torsion modes.
C
      DO 240 I = NB+1,NB+NT
      DO 240 J = I,NB+NT
         KFOAM(I,J) = 2.*QT11FOAM/24.*((0.824*TRATIO*CHORD/
     &      2.)**3-((ZU(1)-ZL(NPLIES))/2.)**3)/3.*CHORD*LENGTH*
     &      INTGRL('X',I,2,J,2,STYLO,1.)/LENGTH**4 + 8.*
     &      QT66FOAM*((0.779*TRA IO*CHORD/2.)**3-((ZU(1)-
     &      ZL(NPLIES))/2.)**3)/3.*CHORD*LENGTH*
     &      INTGRL('X',I,1,J,1,STYLO,1.)/(LENGTH*CHORD)**2
```

250

```fortran
          KFOAM(J,I) = KFOAM(I,J)
240   CONTINUE
C
C     Calculate the contribution to the stiffness matrix from
C     the styrofoam bending-torsion coupling.
C
      DO 250 I = 1,NB
      DO 250 J = MAX0(I,NB+1),NB+NT
          KFOAM(I,J) = -.01585*CHORD*LENGTH*QT11FOAM*(TRATIO*
     &    CHORD)**3*INTGRL('X',I,2,J,2,STYLO,1.)/LENGTH**4
          KFOAM(J,I) = KFOAM(I,J)
250   CONTINUE
C
C     Calculate the contribution to the stiffness matrix from
C     the styrofoam chordwise bending modes.
C
      IF (NC.GT.0) THEN
      DO 260 I = 1,NB+NT+NC
      DO 260 J = MAX0(I,NB+NT+1),NB+NT+NC
          KFOAM(I,J) = 2.*CHORD*LENGTH*((.779*TRATIO*CHORD/
     &    2.)**3-((ZU(1)-ZL(NPLIES))/2.)**3)/3.*(QT11FOAM/
     &    (LENGTH**4)*INTGRL('X',I,2,J,2,STYLO,1.)*
     &    INTGRL('Y',I,0,J,0,-.5,+.5) + QT22FOAM/
     &    (CHORD**4)*INTGRL('X',I,0,J,0,STYLO,1.)*
     &    INTGRL('Y',I,2,J,2,-.5,+.5) + 4.*QT66FOAM/
     &    (LENGTH*CHORD)**2*INTGRL('X',I,1,J,1,STYLO,1.)*
     &    INTGRL('Y',I,1,J,1,-.5,+.5) + QT12FOAM/
     &    (LENGTH*CHORD)**2*(INTGRL('X',I,2,J,0,STYLO,1.)*
     &    INTGRL('Y',I,0,J,2,-.5,+.5)+INTGRL('X',I,0,J,2,
     &    STYLO,1.)*INTGRL('Y',I,2,J,0,-.5,+.5)))
          KFOAM(J,I) = KFOAM(I,J)
260   CONTINUE
      ENDIF
C
C     Calculate the contribution to the stiffness matrix from
C     the styrofoam fore-&-aft modes.
C
      IF (NF.GT.0) THEN
      DO 270 I = 1,NB+NT+NC+NF
      DO 270 J = MAX0(I,NB+NT+NC+1),NB+NT+NC+NF
          IF (I.LE.NB+NT+NC) THEN
              KFOAM(I,J) = 0.
          ELSE
              KFOAM(I,J) = QT11FOAM*(0.685*TRATIO*
     &        CHORD-(ZU(1)-ZL(NPLIES)))*(CHORD**2/12.)*
     &        CHORD*LENGTH*INTGRL('X',I,2,J,2,STYLO,1.)/
     &        LENGTH**4
          ENDIF
          KFOAM(J,I) = KFOAM(I,J)
270   CONTINUE
      ENDIF
C
C     Combine flat plate and styrofoam stiffness and mass matrices
C     to create combines mass and stiffness matrices.
C
      DO 280 I = 1,NMODES
      DO 280 J = 1,NMODES
          M(I,J) = M(I,J)+MFOAM(I,J)
          K(I,J) = K(I,J)+KFOAM(I,J)
280   CONTINUE
      KTT0 = K(NB+1,NB+1)
C
      IF (IOUT.NE.0) THEN
          WRITE(IOUT,'(/A)') ' Styrofoam Mass matrix [kg] : '
```

```
               DO 290 I = 1,NMODES
                  WRITE(IOUT,'(10X,13(1PE10.2))')
     &             (MFOAM(I,J),J=1,NMODES)
  290          CONTINUE
C
               WRITE(IOUT,'(/A)') ' Styrofoam Stiffness matrix [N/m] : '
               DO 300 I = 1,NMODES
                  WRITE(IOUT,'(10X,13(1PE10.2))')
     &             (KFOAM(I,J),J=1,NMODES)
  300          CONTINUE
C
               WRITE(IOUT,'(/A)') ' Combined Total Mass matrix [kg] : '
               DO 310 I = 1,NMODES
                  WRITE(IOUT,'(10X,13(1PE10.2))') (M(I,J),J=1,NMODES)
  310          CONTINUE
C
               WRITE(IOUT,'(/A)') ' Combined Total Stiffness '//
     &          'matrix [N/m] : '
               DO 320 I = 1,NMODES
                  WRITE(IOUT,'(10X,13(1PE10.2))') (K(I,J),J=1,NMODES)
  320          CONTINUE
           ENDIF
C
         ELSEIF ((I1.EQ.2).AND.(TRATIO.LE.0.)) THEN
C
C            Quit on second loop if only flat plate.
C
             GOTO 440
         ENDIF
C
C        Skip static and vibration analyses if no diagnostics requested.
C
         IF (IOUT.EQ.0) GOTO 440
C
C        Calculate plate and/or wing free vibration modes.
C
         IF (I1.EQ.1) WRITE(*,'(A,$)') ' Do plate '
         IF (I1.EQ.2) WRITE(*,'(A,$)') ' Do wing '
         WRITE(*,'(A,$)') 'free vibration analysis ? '
         READ(*,'(A1)') ANSWER
C
         IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) THEN
             IP = 0
             IF (LGEOM) THEN
                 IP = 100
  322            WRITE(*,'(A,I1,A,$)') ' Bending mode (0-',NB,') ? '
                 READ(*,*,ERR=322) NBGEO
                 IF ((NBGEO.LT.0).OR.(NBGEO.GT.NB)) GOTO 322
  325            WRITE(*,'(A,$)') ' Max tip deflection (cm) ? '
                 READ(*,*,ERR=325) WTIPMAX
             ENDIF
C
             IF (I1.EQ.1) WRITE(IOUT,'(/A,$)') ' Plate '
             IF (I1.EQ.2) WRITE(IOUT,'(/A,$)') ' Wing '
             WRITE(IOUT,'(A,$)') 'free vibration freqs'
             IF (LGEOM) WRITE(IOUT,*) ' (LH column = tip deflec in cm):'
             IF (.NOT.LGEOM) WRITE(IOUT,*) ' and mode shapes: '
C
             QGEO = 0.
             DO 345 II = 0,IP
                 IF (IP.NE.0) QGEO = REAL(II)/REAL(IP)*WTIPMAX/100./
     &            FMODE(0,'X',NBGEO,1.)/FMODE(0,'Y',NBGEO,0.)
C
C                Set up dummy mass & stiffness matrices.
```

252

```
C
                DO 330 I = 1,MAXMODE
                DO 330 J = I,MAXMODE
C
C               Add linear contributions.
C
                    MDUM(I,J) = M(I,J)
                    KDUM(I,J) = K(I,J)
C
C               Add nonlinear contributions due to tip deflection.
C
                    IF (LGEOM) THEN
                        IF ((I.GT.NB).AND.(I.LE.NB+NT).AND.
     &                  (J.GT.NB).AND.(J.LE.NB+NT)) THEN
                            KDUM(I,J) = KDUM(I,J) +
     &                      R(NBGEO,NBGEO,I-NB,J-NB)*QGEO**2
                        ELSEIF ((I.GT.NB).AND.(I.LE.NB+NT).AND.
     &                  (J.GT.NB+NT+NC).AND.(J.LE.NB+NT+NC+NF)) THEN
                            KDUM(I,J) = KDUM(I,J) +
     &                      H(NBGEO,I-NB,J-NB-NT-NC)*QGEO
                        ENDIF
                    ENDIF
                    MDUM(J,I) = MDUM(I,J)
                    KDUM(J,I) = KDUM(I,J)
330             CONTINUE
C
C               Call EISPACK eigenvalue solver.
C
                CALL RSG(MAXMODE,NMODES,KDUM,MDUM,FVIB,1,QVIB,
     &          FV1,FV2,IERR)
C
C               Convert to Hertz (negative sign if imaginary).
C
                DO 334 I = 1,NMODES
                    FVIB(I) = SQRT(ABS(FVIB(I)))/(2.*PI)*
     &              (FVIB(I)/ABS(FVIB(I)))
334             CONTINUE
C
C               Print out frequency results.
C
                WRITE(IOUT,'(5X,33F10.2)') QGEO*
     &          FMODE(0,'X',NBGEO,1.)*FMODE(0,'Y',NBGEO,0.)*100.,
     &          (FVIB(I),I=1,NMODES)
                IF ((.NOT.LGEOM).OR.(II.EQ.IP)) THEN
                    DO 340 I = 1,NMODES
                        QMAX = QVIB(1,I)
                        DO 335 J = 1,NMODES
                            IF (ABS(QVIB(J,I)).GT.ABS(QMAX))
     &                      QMAX=QVIB(J,I)
335                     CONTINUE
                        WRITE(IOUT,'(5X,I2,A,F9.2,A,SP,32F7.3)')
     &                  I,' Freq =',FVIB(I),' Hz ; mode shape = ',
     &                  (QVIB(J,I)/QMAX,J=1,NMODES)
340                 CONTINUE
C
                IF ((LGEOM).AND.(II.EQ.IP)) THEN
                DO 342 I = 1,MAXMODE
                DO 342 J = I,MAXMODE
                    MDUM(I,J) = M(I,J)
                    KDUM(I,J) = K(I,J)
                    IF ((I.GT.NB).AND.(I.LE.NB+NT).AND.
     &              (J.GT.NB).AND.(J.LE.NB+NT)) THEN
                        KDUM(I,J) = KDUM(I,J) +
     &                  R(NBGEO,NBGEO,I-NB,J-NB)*QGEO**2
```

253

```
                    ELSEIF ((I.GT.NB).AND.(I.LE.NB+NT).AND.
     &                (J.GT.NB+NT+NC).AND.(J.LE.NB+NT+NC+NF)) THEN
                       KDUM(I,J) = KDUM(I,J) +
     &                   H(NBGEO,I-NB,J-NB-NT-NC)*QGEO
                    ENDIF
                    MDUM(J,I) = MDUM(I,J)
                    KDUM(J,I) = KDUM(I,J)
 342          CONTINUE
              WRITE(IOUT,*) 'Mass matrix: '
              DO 344 I = 1,NMODES
                 WRITE(IOUT,'(99(1PE10.2))') (MDUM(I,J),J=1,NMODES)
 344          CONTINUE
              WRITE(IOUT,*) 'Stiffness matrix: '
              DO 346 I = 1,NMODES
                 WRITE(IOUT,'(99(1PE10.2))') (KDUM(I,J),J=1,NMODES)
 346          CONTINUE
              ENDIF
C
           ENDIF
 345    CONTINUE
        ENDIF
C
C       Loop through tip force (I2=1) and tip moment (I2=2) analyses.
C
        DO 440 I2 = 1,2
C
           IF (I1.EQ.1) WRITE(*,'(A,$)') ' Do plate tip '
           IF (I1.EQ.2) WRITE(*,'(A,$)') ' Do wing tip '
           IF (I2.EQ.1) WRITE(*,'(A,$)') 'force analysis ? '
           IF (I2.EQ.2) WRITE(*,'(A,$)') 'moment analysis ? '
           READ(*,'(A1)') ANSWER
           IF ((ANSWER.EQ.'N').OR.(ANSWER.EQ.'n')) GOTO 440
C
C          Do direct inversion of K matrix if no structural
C          nonlinearities.
C
           IF ((.NOT.LGEOM).AND.(KTTCUBE.EQ.0.)) THEN
              K(NB+1,NB+1) = KTT0
              FORCE = 1.
              DO 347 I = 1,NB+NT+NC
                 QLIT(I) = 0.
                 IF (I2.EQ.1) RES(I) = FMODE(0,'X',I,1.)*
     &             FMODE(0,'Y',I,0.)*FORCE
                 IF (I2.EQ.2) RES(I) = FMODE(0,'X',I,1.)*
     &             FMODE(1,'Y',I,0.)*FORCE/CHORD
 347          CONTINUE
C
              CALL SOLVE(K,RES,QLIT,MAXMODE,1,(NB+NT+NC))
C
              DEFLEC = 0.
              TWIST = 0.
              DO 348 I = 1,NB+NT+NC
                 DEFLEC = DEFLEC + QLIT(I)*FMODE(0,'X',I,1.)*
     &             FMODE(0,'Y',I,0.)
                 TWIST = TWIST + QLIT(I)/CHORD*FMODE(0,'X',I,1.)*
     &             FMODE(1,'Y',I,0.)
 348          CONTINUE
              DEFLEC = DEFLEC*100.
              TWIST = ATAN(TWIST)*180./PI
              IF (I2.EQ.1) THEN
                 WRITE(IOUT,*) ' F/h =',FORCE/DEFLEC,' N/m'
                 WRITE(IOUT,*) ' F/a =',FORCE/TWIST,' N/deg'
              ELSEIF (I2.EQ.2) THEN
                 WRITE(IOUT,*) ' M/h =',FORCE/DEFLEC,' Nm/m'
```

```
                    WRITE(IOUT,*) ' M/a =',FORCE/TWIST,' Nm/deg'
                 ENDIF
            ELSE
C
C           Read in concentrated flat plate force data.
C
            IF ((I1.EQ.1).AND.(I2.EQ.1)) FILENAME =
     &       TRIM(LAYUP)//'PNXS.FOR '
            IF ((I1.EQ.1).AND.(I2.EQ.2)) FILENAME =
     &       TRIM(LAYUP)//'PNXS.MOM '
            IF ((I1.EQ.2).AND.(I2.EQ.1)) FILENAME =
     &       TRIM(LAYUP)//'WNXS.FOR '
            IF ((I1.EQ.2).AND.(I2.EQ.2)) FILENAME =
     &       TRIM(LAYUP)//'WNXS.MOM '
            OPEN(3,FILE=TRIM(FILENAME),STATUS='OLD',
     &       FORM='FORMATTED',ERR=360)
            IF (I2.EQ.1) WRITE(*,'(2A/)') ' Force data taken from ',
     &       TRIM(FILENAME)
            IF (I2.EQ.2) WRITE(*,'(2A/)') ' Moment data taken from ',
     &       TRIM(FILENAME)
            IP = 0
            XLO = 0.
            XHI = 0.
C
C           Read in tip force (N) or moment (Nm), tip deflec (cm), and
C           tip twist (deg), and determine LO and HI.
C
350         READ(3,*,END=360) X,Y1,Y2
            IF (X.GT.XHI) XHI=X
            IF (X.LT.XLO) XLO=X
            IP = IP+1
            GOTO 350
360         CLOSE(3)
C
            IF ((I1.EQ.1).AND.(I2.EQ.1)) FILENAME =
     &       TRIM(LAYUP)//'PNAS.FOR '
            IF ((I1.EQ.1).AND.(I2.EQ.2)) FILENAME =
     &       TRIM(LAYUP)//'PNAS.MOM '
            IF ((I1.EQ.2).AND.(I2.EQ.1)) FILENAME =
     &       TRIM(LAYUP)//'WNAS.FOR '
            IF ((I1.EQ.2).AND.(I2.EQ.2)) FILENAME =
     &       TRIM(LAYUP)//'WNAS.MOM '
            OPEN(3,FILE=TRIM(FILENAME),STATUS='NEW',
     &       FORM='FORMATTED',ERR=440)
C
            XLO = 2.*XLO
            XHI = 2.*XHI
            IIMAX = IP*8
            DO 430 II = 1,IIMAX
C
C              Calculate current force/moment value.
C
               FORCE = XLO + (XHI-XLO)*REAL(II-1)/REAL(IIMAX-1)
               IF (I2.EQ.1) WRITE(*,'(A,F9.3,A)') ' Calculating '//
     &          'for force =',FORCE,' N'
               IF (I2.EQ.2) WRITE(*,'(A,F9.3,A)') ' Calculating '//
     &          'for moment =',FORCE,' Nm'
C
C              Initialize modal amplitudes and residuals.
C
               K(NB+1,NB+1) = KTT0
               DO 370 I = 1,NB+NT+NC
                  QLIT(I) = 0.
                  IF (I2.EQ.1) RES(I) = FMODE(0,'X',I,1.)*
```

255

```fortran
     &                     FMODE(0,'Y',I,0.)*FORCE
                     IF (I2.EQ.2) RES(I) = FMODE(0,'X',I,1.)*
     &                     FMODE(1,'Y',I,0.)*FORCE/CHORD
370               CONTINUE
C
C                 Loop through Newton-Raphson solver until converged.
C
                  CONVERGED = .FALSE.
                  DO WHILE (.NOT.CONVERGED)
C
C                     Calculate derivative matrix.
C
                     DO 380 I = 1,NB+NT+NC
                     DO 380 J = 1,NB+NT+NC
                        DRDQ(I,J) = -K(I,J)
380                  CONTINUE
                     DRDQ(NB+1,NB+1) = DRDQ(NB+1,NB+1)-2.*KTTCUBE*
     &                  QLIT(NB+1)**2
C
C                     Apply Newton-Raphson step.
C
                     CALL SOLVE(DRDQ,RES,DQ,MAXMODE,1,NB+NT+NC)
                     DO 390 I = 1,NB+NT+NC
                        QLIT(I) = QLIT(I)-DQ(I)
390                  CONTINUE
C
C                     Calculate new residuals and test for convergence.
C
                     K(NB+1,NB+1) = KTT0+KTTCUBE*QLIT(NB+1)**2
                     CONVERGED = .TRUE.
                     DO 410 I = 1,NB+NT+NC
                        IF (I2.EQ.1) RES(I) = FMODE(0,'X',I,1.)*
     &                     FMODE(0,'Y',I,0.)*FORCE
                        IF (I2.EQ.2) RES(I) = FMODE(0,'X',I,1.)*
     &                     FMODE(1,'Y',I,0.)*FORCE/CHORD
                        DO 400 J = 1,NB+NT+NC
                           RES(I) = RES(I) - K(I,J)*QLIT(J)
400                     CONTINUE
                        IF (ABS(RES(I)).GT.1.E-6) CONVERGED=.FALSE.
410                  CONTINUE
C
                  END DO
C
C                 Calculate tip deflection (in cm) and tip twist (in deg).
C
                  DEFLEC = 0.
                  TWIST = 0.
                  DO 420 I = 1,NB+NT+NC
                     DEFLEC = DEFLEC + QLIT(I)*FMODE(0,'X',I,1.)*
     &                  FMODE(0,'Y',I,0.)
                     TWIST = TWIST + QLIT(I)/CHORD*FMODE(0,'X',I,1.)*
     &                  FMODE(1,'Y',I,0.)
420               CONTINUE
                  DEFLEC = DEFLEC*100.
                  TWIST = ATAN(TWIST)*180./PI
                  WRITE(3,'(3(1PE12.4))') FORCE,DEFLEC,TWIST
C
430            CONTINUE
               CLOSE(3)
            ENDIF
C
440      CONTINUE
         CLOSE(2)
C
```

```fortran
999     RETURN
        END

C-----FILE: STIFF.FOR -------------------------------------------------
C
        SUBROUTINE QUCON(EL,ET,NULT,GLT,QU11,QU12,QU22,QU66)
C
C       Subroutine to compute the unidirectional elastic constants, the
C       uni-directional Q's, from the ply engineering elastic constants
C       (all in Pa).
C
        REAL    EL,ET,NULT,NUTL,GLT
        REAL    QU11,QU12,QU22,QU66
C
        NUTL = ET/EL*NULT
        DENOM = 1.-NULT*NUTL
        QU11 = EL/DENOM
        QU12 = NULT*ET/DENOM
        QU22 = ET/DENOM
        QU66 = GLT
        RETURN
        END
C
C=====================================================================
C
        SUBROUTINE QTCON(K,THETA,QU11,QU12,QU22,QU66,QT)
C
C       Subroutine to compute the rotated elastic constants, the Q[theta]
C       (Pa), for the K-th ply, laid up at an angle theta (rad).
C
        INTEGER         K
        REAL            THETA,QU11,QU12,QU22,QU66
        REAL            I1,I2,R1,R2,QT(3,3,*)
C
C       Calculate the invariants
C
        I1 = (QU11 + QU22 + 2.*QU12)/4.
        I2 = (QU11 + QU22 - 2.*QU12 + 4.*QU66)/8.
        R1 = (QU11 - QU22)/2.
        R2 = (QU11 + QU22 - 2.*QU12 - 4.*QU66)/8.
C
        QT(1,1,K) = I1 + I2 + R1*COS(2.*THETA) + R2*COS(4.*THETA)
        QT(2,2,K) = I1 + I2 - R1*COS(2.*THETA) + R2*COS(4.*THETA)
        QT(1,2,K) = I1-I2 - R2*COS(4.*THETA)
        QT(3,3,K) = I2 - R2*COS(4.*THETA)
        QT(1,3,K) = R1*SIN(2.*THETA)/2. + R2*SIN(4.*THETA)
        QT(2,3,K) = R1*SIN(2.*THETA)/2. - R2*SIN(4.*THETA)
        QT(2,1,K) = QT(1,2,K)
        QT(3,1,K) = QT(1,3,K)
        QT(3,2,K) = QT(2,3,K)
C
        RETURN
        END
C
C=====================================================================
C
        SUBROUTINE BEND(NPLIES,ZU,ZL,QT,A,D)
C
C       Subroutine to compute the laminate bending stiffnesses,
C       Aij (N/m) & Dij (N-m).
C
        INTEGER         NPLIES
        REAL            ZU(*),ZL(*),QT(3,3,*)
        REAL            A(3,3),D(3,3)
C
```

```
C       Initialize the A and D matrices.
C
        DO 10 I = 1,3
        DO 10 J = 1,3
           A(I,J) = 0.
           D(I,J) = 0.
10      CONTINUE
C
C       Add the contribution of each ply to the A & D matrices.
C
        DO 30 I = 1,3
        DO 30 J = I,3
        DO 30 K = 1,NPLIES
           A(I,J) = A(I,J) + QT(I,J,K)*(ZU(K)-ZL(K))
           D(I,J) = D(I,J) + QT(I,J,K)*(ZU(K)**3-ZL(K)**3)/3.
30      CONTINUE
C
        RETURN
        END
C
C=======================================================================
C
        SUBROUTINE STIFF(A,D,LO,HI)
C
C       Subroutine to compute the stiffness matrix, Kij (N/m).
C
        INCLUDE     PARAM.INC
        INCLUDE     GLBBLK.INC
        REAL        A(3,3),D(3,3),LO,HI,INTGRL
C
C       NOTE: INTGRL(XY,I,ID,J,JD,lo,hi) is the function to numerically
C       integrate the XY-variation of the ID-th derivative of the I-th
C       mode with the JD-th derivative of the J-th mode between the
C       interval [lo,hi].
C
        DO 10 I = 1,NB+NT+NC
        DO 10 J = I,NB+NT+NC
C
C          Calculate out-of-plane stiffness matrix components.
C
           K(I,J) = (CHORD*LENGTH) * (D(1,1)*INTGRL('X',I,2,J,2,LO,HI)*
     &      INTGRL('Y',I,0,J,0,-.5,+.5)/LENGTH**4 + D(2,2)*INTGRL('X',
     &      I,0,J,0,LO,HI)*INTGRL('Y',I,2,J,2,-.5,+.5)/CHORD**4 + 4.*
     &      D(3,3)*INTGRL('X',I,1,J,1,LO,HI)*INTGRL('Y',I,1,J,1,-.5,
     &      +.5)/(LENGTH*CHORD)**2 + D(1,2)*(INTGRL('X',I,2,J,0,LO,HI)*
     &      INTGRL('Y',I,0,J,2,-.5,+.5)+INTGRL('X',I,0,J,2,LO,HI)*
     &      INTGRL('Y',I,2,J,0,-.5,+.5))/(LENGTH*CHORD)**2 + 2.*
     &      D(1,3)*(INTGRL('X',I,2,J,1,LO,HI)*INTGRL('Y',I,0,J,1,-.5,
     &      +.5)+INTGRL('X',I,1,J,2,LO,HI)*INTGRL('Y',I,1,J,0,-.5,+.5))/
     &      (LENGTH**3*CHORD) + 2.*D(2,3)*(INTGRL('X',I,0,J,1,LO,HI)*
     &      INTGRL('Y',I,2,J,1,-.5,+.5)+INTGRL('X',I,1,J,0,LO,HI)*
     &      INTGRL('Y',I,1,J,2,-.5,+.5))/(LENGTH*CHORD**3))
           K(J,I) = K(I,J)
10      CONTINUE
C
        IF (NF.GT.0) THEN
        DO 20 I = 1,NB+NT+NC+NF
        DO 20 J = MAX0(I,NB+NT+NC+1),NB+NT+NC+NF
C
C       Calculate fore-&-aft stiffness matrix components.
C
        IF (I.LE.NB+NT+NC) THEN
           K(I,J) = 0.
        ELSE
```

258

```fortran
      K(I,J) = (CHORD*LENGTH) * (A(1,1)*INTGRL('X',I,2,J,2,LO,HI)*
     & INTGRL('Y',I,0,J,0,-.5,+.5)/LENGTH**4 + A(2,2)*INTGRL('X',
     & I,0,J,0,LO,HI)*INTGRL('Y',I,2,J,2,-.5,+.5)/CHORD**4 + 4.*
     & A(3,3)*INTGRL('X',I,1,J,1,LO,HI)*INTGRL('Y',I,1,J,1,-.5,
     & +.5)/(LENGTH*CHORD)**2 + A(1,2)*(INTGRL('X',I,2,J,0,LO,HI)*
     & INTGRL('Y',I,0,J,2,-.5,+.5)+INTGRL('X',I,0,J,2,LO,HI)*
     & INTGRL('Y',I,2,J,0,-.5,+.5))/(LENGTH*CHORD)**2 + 2.*
     & A(1,3)*(INTGRL('X',I,2,J,1,LO,HI)*INTGRL('Y',I,0,J,1,-.5,
     & +.5)+INTGRL('X',I,1,J,2,LO,HI)*INTGRL('Y',I,1,J,0,-.5,+.5))/
     & (LENGTH**3*CHORD) + 2.*A(2,3)*(INTGRL('X',I,0,J,1,LO,HI)*
     & INTGRL('Y',I,2,J,1,-.5,+.5)+INTGRL('X',I,1,J,0,LO,HI)*
     & INTGRL('Y',I,1,J,2,-.5,+.5))/(LENGTH*CHORD**3)) *
     & (CHORD**2/12.)
      K(I,J) = (CHORD*LENGTH) * A(1,1)*INTGRL('X',I,2,J,2,LO,HI)*
     & INTGRL('Y',I,0,J,0,-.5,+.5)/LENGTH**4 * (CHORD**2/12.)
      ENDIF
      K(J,I) = K(I,J)
C
20    CONTINUE
      ENDIF
C

      RETURN
      END
```

259

```
C-----FILE: PARAM.INC --------------------------------------------------
C
C       "Include" file, PARAM.INC, which describes the general
C       parameters of the stall flutter analysis programs.
C
        PARAMETER    (PI=3.141592653589793238)
        PARAMETER    (RHOA=1.226)
        PARAMETER    (RMUA=1.78E-5)
        PARAMETER    (MAXPLIES=20)
        PARAMETER    (NBMAX=10)
        PARAMETER    (NTMAX=10)
        PARAMETER    (NCMAX=2)
        PARAMETER    (NFMAX=10)
        PARAMETER    (MAXMODE=32)
        PARAMETER    (MAXREG=5)
        PARAMETER    (MAXPOW=20)
C
C       RHOA:       Air density in kg/m**3
C       RMUA:       Air coefficient of viscocity in kg/m-sec
C       MAXPLIES:   Maximum allowable number of plies in analysis
C       NBMAX:      Maximum allowable number of bending modes in analysis
C       NTMAX:      Maximum allowable number of torsion modes in analysis
C       NCMAX:      Maximum allowable number of chordwise bending modes
C       NFMAX:      Maximum allowable number of fore/aft modes in analysis
C       MAXMODE:    Maximum allowable number of mode shapes in analysis
C       MAXREG:     Maximum allowable number of describing regions
C                   for aerodynamic force curves
C       MAXPOW:     Maximum allowable polynomial power for each
C                   describing region for aerodynamic force curves
C


C-----FILE: GLBBLK.INC -------------------------------------------------
C
C       "Include" file to describe variables used globally
C       by most programs in the stall flutter analysis.
C
        REAL         RE,CHORD,LENGTH,LAMBDA,KTT0,KTTCUBE
        REAL         M(MAXMODE,MAXMODE),K(MAXMODE,MAXMODE)
        REAL         MDUM(MAXMODE,MAXMODE),KDUM(MAXMODE,MAXMODE)
        REAL         QVIB(MAXMODE,MAXMODE),FVIB(MAXMODE)
        REAL         FV1(MAXMODE),FV2(MAXMODE)
        REAL         H(0:NBMAX,NTMAX,NFMAX)
        REAL         R(0:NBMAX,0:NBMAX,NTMAX,NTMAX)
        REAL         BETA,KT(NTMAX),G(NTMAX),F(NTMAX),B(NTMAX,4)
        INTEGER      NB,NT,NC,NF,NMODES,ATYPE
        LOGICAL      LINEAR,STEADY,REDUC,CORREC
        LOGICAL      VLINES,LATAN,LAEROF,LCONST,LGEOM
        CHARACTER    FOIL*5
        CHARACTER*2  MLABEL(MAXMODE)
        COMMON       RE,CHORD,LENGTH,LAMBDA,KTT0,KTTCUBE,M,K
        COMMON       MDUM,KDUM,QVIB,FVIB,FV1,FV2,H,R
        COMMON       BETA,KT,G,F,B,NB,NT,NC,NF,NMODES
        COMMON       ATYPE,LINEAR,STEADY,REDUC,CORREC,VLINES
        COMMON       LATAN,LAEROF,LCONST,LGEOM,FOIL,MLABEL
C
C       RE:         Reynold's number (non-dim)
C       CHORD:      Chord length (m)
C       LENGTH:     Half-span (m)
C       LAMBDA:     Sweep angle (deg)
C       KTT0:       Torsional linear term (N/m)
C       KTTCUBE:    Torsional cubic factor (1/m**2)
C       M(i,j):     Mass matrix (kg)
C       K(i,j):     Stiffness matrix (N/m)
C       BETA,KT,G,F,B:   Coefficients of torsional mode shapes
```

260

```
C     NB:          Number of bending modes in analysis
C     NT:          Number of torsion modes in analysis
C     NF:          Number of fore-aft modes in analysis
C     NC:          Number of chordwise bending modes in analysis
C     NMODES:      Total number of modes in analysis
C     ATYPE:       Type of aerodynamic analysis to use [see AEROF.FOR]
C     LINEAR:      Logical variable, if linear analysis is to be done
C     STEADY:      Logical variable, if steady analysis is to be done
C     REDUC:       Logical variable to tell if finite-span reduction is
C                  to be applied to aerodynamic forces
C     CORREC:      Logical variable to tell if spanwise correction is to
C                  be applied to spanwise integrations
C     VLINES:      Logical variable to tell if constant velocity lines or
C                  constant angle lines are to be calculated by analysis
C     LATAN:       Logical variable to tell if exact angle or small-
C                  angle-approximations are to be applied to angle
C                  calculations
C     LAEROF:      Logical variable to tell whether to print diagnotics
C                  each time the AEROF unsteady aerodynamics subroutine
C                  is called
C     LCONST:      Logical variable to tell whether to use constant
C                  coefficients in unsteady aerodynamic analysis
C     FOIL:        Character variable that denotes airfoil type
C

C----- FILE: GAUSS.INC ------------------------------------------------
C
      INTEGER      GPOINTS
      PARAMETER    (GPOINTS=12)
      REAL         GP(GPOINTS),GW(GPOINTS)
      DATA         GP/-.981560634,-.904117256,-.769902674,
     &                -.587317954,-.367831499,-.125233409,
     &                .125233409,.367831499,.587317954,
     &                .769902674,.904117256,.981560634/
      DATA         GW/.047175336,.106939326,.160078329,
     &                .203167427,.233492537,.249147046,
     &                .249147046,.233492537,.203167427,
     &                .160078329,.106939326,.047175336/
```
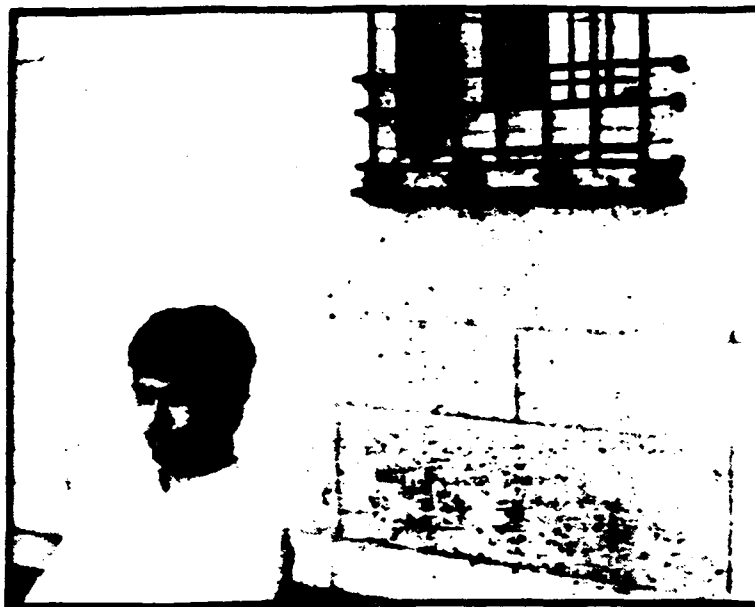
Photo courtesy Christopher J. Andrews

Biography

The author, Peter Earl Dunn, was born on July 13, 1963, in Montreal, Quebec, the firstborn child of Drs. Earl Vincent and Ruth Cleto Dunn. He is a Canadian citizen. He attended Grades K through 13 at the Toronto French School, and is fluent in both French and English. He attended the Massachusetts Institute of Technology as an undergraduate from Sept. 1981 to June 1985, completing two Bachelor of Science degrees in Aeronautics & Astronautics and in Applied Mathematics, with a minor in Film & Media Studies. During his junior & senior years as an undergraduate at MIT, he worked as teaching assistant for the Aero & Astro Department's Unified Engineering Class. Upon graduation he was awarded the Henry Webb Salisbury award for scholarship from the MIT Aero & Astro Department.

He has been a graduate student at MIT from Sept. 1985 to Dec. 1991. His first year as a graduate student, he worked on modal analysis of the aerodynamics of turbine rows for Prof. Edward Crawley under the auspices of the MIT Lester B. Gardner Fellowship. Since Sept. 1986 he has worked as a research assistant at MIT's Technology Laboratory for Advanced Composites, under the supervision of Prof. John Dugundji, with the funding of the Air Force Office of Scientific Research. He received his Master's degree from MIT in June 1989 — his thesis topic was a preliminary study for the work of his doctoral dissertation. He presented part of this work at the 31st Structures, Structural Dynamics and Materials Conference in Long Beach, Calif. during April 1990.

He is a member of the AIAA, and the honor societies of Sigma Gamma Tau, Tau Beta Pi, Phi Beta Kappa, and Sigma Xi. Since 1985 he has served as a board member of *The Tech*, the MIT campus newspaper.